

Appln. No. 10/729,007
G.AU:NYA日本国特許庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日 2003年 6月20日
Date of Application:

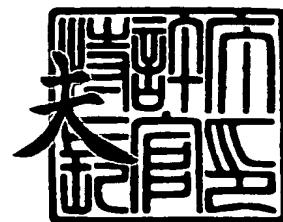
出願番号 特願2003-176930
Application Number:
[ST. 10/C]: [JP 2003-176930]

出願人 キヤノン株式会社
Applicant(s):

2004年 1月 6日

特許庁長官
Commissioner,
Japan Patent Office

今井 康



【書類名】 特許願

【整理番号】 253523

【提出日】 平成15年 6月20日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 7/00

【発明の名称】 画像処理方法、画像処理装置

【請求項の数】 7

【発明者】

【住所又は居所】 東京都大田区下丸子3丁目30番2号 キヤノン株式会社
社内

【氏名】 石川 智恵

【特許出願人】

【識別番号】 000001007

【氏名又は名称】 キヤノン株式会社

【代理人】

【識別番号】 100076428

【弁理士】

【氏名又は名称】 大塚 康德

【電話番号】 03-5276-3241

【選任した代理人】

【識別番号】 100112508

【弁理士】

【氏名又は名称】 高柳 司郎

【電話番号】 03-5276-3241

【選任した代理人】

【識別番号】 100115071

【弁理士】

【氏名又は名称】 大塚 康弘

【電話番号】 03-5276-3241

【選任した代理人】

【識別番号】 100116894

【弁理士】

【氏名又は名称】 木村 秀二

【電話番号】 03-5276-3241

【手数料の表示】

【予納台帳番号】 003458

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 0102485

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 画像処理方法、画像処理装置

【特許請求の範囲】

【請求項 1】 複数の論理単位から成るタイル単位で分割された画像の符号化データで、かつ、当該符号化データのフォーマットが論理単位の並べる順番が複数種類可能である符号化データを保持する装置から、各タイルについて所望の画像を得るために必要な数の論理単位のデータを受信する画像処理装置が行う画像処理方法であって、

受信する各論理単位を管理するための管理情報を作成する管理情報作成工程と

、
受信した各々の論理単位を、夫々の論理単位が属するタイルの番号に従って並び替え、更に、同じタイルに属する夫々の論理単位を当該タイル中の順番に従って並び替える並び替え工程と、

前記並び替え工程で並び替えた論理単位のデータを前記管理情報に順次付加すると共に、付加した論理単位のデータと前記管理情報とを含むキャッシュデータ中の、各論理単位のデータの配置に関する情報を前記管理情報に登録する登録工程と

を備えることを特徴とする画像処理方法。

【請求項 2】 受信した論理単位が属するタイルの番号、及び当該論理単位の当該タイル中の順番は、論理単位を受信に先立って受信したヘッダ情報、及び受信した論理単位のデータを解析することにより得られることを特徴とする請求項 1 に記載の画像処理方法。

【請求項 3】 前記管理情報作成工程では、各タイルについて、J P E G 2 0 0 0 のプログレッションオーダーに従った論理単位の並びを管理する為の管理情報を作成することを特徴とする請求項 1 に記載の画像処理方法。

【請求項 4】 前記登録工程では更に、前記並べ替え工程で並べ替えた夫々の論理単位が、夫々の論理単位が属するタイル内で連続した順番であることを示す情報を前記管理情報に登録することを特徴とする請求項 1 に記載の画像処理方法。

。

【請求項 5】 複数の論理単位から成るタイル単位で分割された画像の符号化データで、かつ、当該符号化データのフォーマットが論理単位の並べる順番が複数種類可能である符号化データを保持する装置から、各タイルについて所望の画像を得るために必要な数の論理単位のデータを受信する画像処理装置であって、

受信する各論理単位を管理するための管理情報を作成する管理情報作成手段と、

受信した各々の論理単位を、夫々の論理単位が属するタイルの番号に従って並び替え、更に、同じタイルに属する夫々の論理単位を当該タイル中の順番に従って並び替える並び替え手段と、

前記並び替え手段が並び替えた論理単位のデータを前記管理情報に順次付加すると共に、付加した論理単位のデータと前記管理情報とを含むキャッシュデータ中の、各論理単位のデータの配置に関する情報を前記管理情報に登録する登録手段と

を備えることを特徴とする画像処理装置。

【請求項 6】 コンピュータに請求項 1 乃至 4 の何れか 1 項に記載の画像処理方法を実行させるためのプログラム。

【請求項 7】 請求項 6 に記載のプログラムを格納するコンピュータ読み取り可能な記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、複数のパケットから成るタイル単位で分割された画像の符号化データを保持する装置から、各タイルについて所望の画像を得るために必要な数のパケットのデータを受信し、これを管理するための技術に関するものである。

【0002】

【従来技術】

インターネット上では、クライアント端末装置上で動作する Web ブラウザを用いて WWW サーバにアクセスし、Web ブラウザを介してクライアント端末装

置側で文書データや画像データ等を閲覧することが盛んに行われている。このWWWサーバは、ホームページといわれる公開したい情報をHTMLで記述した情報を保持しており、それをクライアント端末装置側のWebブラウザが解釈して、クライアント端末装置が有する表示装置の表示画面上に表示する。クライアント端末装置側のWebブラウザは、表示装置に表示されているホームページ中のリンク部分がユーザによって指示されると、このリンク部分が示すリンク先を辿って行き、必要な情報を得ることができる。

【 0 0 0 3 】

さらに、サーバが管理しているファイルをダウンロードする方法として、File Transfer Protocol（以下FTPと略す）という方法がある。このFTPとは、ネットワークを通して、サーバ上にあるファイルを一度にクライアント端末装置に転送する仕組みである。

【 0 0 0 4 】

あるいは、画像データファイルへ断片的にアクセスして表示するためのプロトコルとして、Flashpix/IIPがある。このIIP(Internet Imaging Protocol)は、Flashpixという画像データファイルフォーマットに最適なプロトコルになっており、画像データの部分アクセスの単位はFlashpixのタイル単位に行うものとなっている。このIIPプロトコルを使用する技術としては従来からいくつかが開示されている（例えば特許文献1を参照）。

【 0 0 0 5 】

これらのいずれのプロトコルであっても、サーバから送信されるデータは、それぞれ独立した符号データである。そのため、サーバ側では、送信データの順番を考慮して返送する必要がなかった。

【 0 0 0 6 】

また、J P E G 2 0 0 0に従って符号化されたファイルへ断片的にアクセスして表示するためのプロトコルとして、JPEG2000 image coding system - Part 9: Interactivity tools, APIs and protocols(以下、JPIPと記す)が検討されている。このようなプロトコルを使ってJPEG2000の画像データをクライアント端末装置に断片的に送信した場合、送信された画像データをクライアント端末装置がデ

コードするためには、受信した断片的な符号化データを1つのファイルにキャッシュする必要がある。これは、JPEG2000の各スケーラビリティの符号化データは、そのスケーラビリティより1つ下のスケーラビリティのデータからの差分データであるためである。

【0007】

このように断片的に受信したJPEG2000データを1つのファイルにキャッシュする際に、受信した順番にデータをキャッシュファイルにアペンドする方法がある。この場合、簡単にひとつのファイルにキャッシュデータを書き込むことができる。その際、キャッシュされたデータを管理するために、各データの識別データとオフセット値及びデータ長を書き込んでいた。

【0008】

このJPIPを利用してデータを要求するクライアントは、レスポンスデータの返送方法に関して、次の2種類の要求が可能になっている。

【0009】

1) fullwindow: レスポンスデータの送信順序はサーバに一任する変わりに、要求データを全て確実に返送することを要求

2) progressive: 要求データの一部を削減しても良いから、画質方向のスケーラビリティ(layer progression order)で返送することを要求

従って、クライアントは、要求したデータの全てを確実に受信したい場合には、上記のprogressiveでの要求をおこうため、サーバは、クライアント側の処理を考慮せずに、サーバの実装方針に従って、返送していた。

【0010】

【特許文献1】

特開 2002-49514

【0011】

【発明が解決しようとする課題】

しかし、サーバから送られてくるパケットデータは、必ずしもサーバにある画像のオリジナルのprogression orderの順番ではない。また、クライアント端末装置が複数のスレッドを立ち上げて、1つのJPEG2000ファイルから一度に複数の

スレッドでパケットデータを取得した場合には、送られてくるパケットデータは progression order の順番である可能性は、ますます小さくなる。

【0012】

そのため、パケットデータを送られてきた順番どおりにキャッシュファイルに書き込むと、キャッシュされたデータを管理する情報へのアクセスがランダムになり、キャッシュデータを書き込む際にキャッシュファイル内のシーク回数が多くなり、時間がかかってしまうという問題が発生する。

【0013】

さらに、このようにJPEG2000の断片的なデータを受信したクライアント端末装置が受信したデータをデコードし、表示する時には、キャッシュデータからデコードが処理できるデータの順番にキャッシュデータを並べ直して、読み出す必要がある。例えば、JPEG2000の汎用的なデコードを利用する場合には、JPEG2000ビットストリームシンタックスに沿ってキャッシュデータを読み出して、一本のビットストリームを作成する。したがって、従来の方法のように、受信した順番にデータをアペンドしていくと、キャッシュデータを読み出す際にも、キャッシュファイル内のシーク回数が多くなり、時間がかかってしまう問題が発生する。

【0014】

本発明は以上の問題に鑑みて成されたものであり、符号化データを受信してキャッシュする場合に、キャッシュしたデータに対してアクセスする際に要する時間や回数を軽減させる様な形態でキャッシュを行うことを目的とする。

【0015】

【課題を解決するための手段】

本発明の目的を達成するために、例えば本発明の画像処理方法は以下の構成を備える。

【0016】

すなわち、複数の論理単位から成るタイル単位で分割された画像の符号化データで、かつ、当該符号化データのフォーマットが論理単位の並べる順番が複数種類可能である符号化データを保持する装置から、各タイルについて所望の画像を得るために必要な数の論理単位のデータを受信する画像処理装置が行う画像処理

方法であって、

受信する各論理単位を管理するための管理情報を作成する管理情報作成工程と

、
受信した各々の論理単位を、夫々の論理単位が属するタイルの番号に従って並び替え、更に、同じタイルに属する夫々の論理単位を当該タイル中の順番に従って並び替える並び替え工程と、

前記並び替え工程で並び替えた論理単位データを前記管理情報に順次付加すると共に、付加した論理単位データと前記管理情報とを含むキャッシュデータ中の、各論理単位データの配置に関する情報を前記管理情報に登録する登録工程と

を備えることを特徴とする。

【0017】

本発明の目的を達成するために、例えば本発明の画像処理装置は以下の構成を備える。

【0018】

すなわち、複数の論理単位から成るタイル単位で分割された画像の符号化データで、かつ、当該符号化データのフォーマットが論理単位の並べる順番が複数種類可能である符号化データを保持する装置から、各タイルについて所望の画像を得るために必要な数の論理単位データを受信する画像処理装置であって、

受信する各論理単位を管理するための管理情報を作成する管理情報作成手段と

、
受信した各々の論理単位を、夫々の論理単位が属するタイルの番号に従って並び替え、更に、同じタイルに属する夫々の論理単位を当該タイル中の順番に従って並び替える並び替え手段と、

前記並び替え手段が並び替えた論理単位データを前記管理情報に順次付加すると共に、付加した論理単位データと前記管理情報とを含むキャッシュデータ中の、各論理単位データの配置に関する情報を前記管理情報に登録する登録手段と

を備えることを特徴とする。

【0019】

【発明の実施の形態】

以下添付図面を参照して、本発明を好適な実施形態に従って詳細に説明する。

【0020】

[第1の実施形態]

図1はPC（パーソナルコンピュータ）やワークステーションなどにより構成される、本実施形態に係る画像処理装置の基本構成を示すブロック図である。

【0021】

CPU101は一次記憶メモリ102にロードされたプログラムやデータを用いて本装置全体の制御を行うと共に後述の各部の動作の制御を行う。

【0022】

一次記憶メモリ102はRAMに代表される主記憶装置であって、二次記憶メモリ103からロードされたプログラムやデータを一時的に記憶するエリアを備えると共に、CPU101が各種の処理を行うために使用するエリアを備える。

【0023】

二次記憶メモリ103は、ハードディスクドライブに代表される大記憶容量を有する情報記憶装置であって、ここにOS（オペレーティングシステム）や後述の各処理をCPU101が実行するためのプログラムやデータを保存しておくことができ、CPU101による制御によって、要求されたプログラムやデータを一次記憶メモリ102に出力（ロード）する。

【0024】

入力デバイス104は、例えばマウスやキーボードなどに代表される指示入力装置であって、この入力デバイス104により、本装置のユーザは各種の指示をCPU101に入力することができる。

【0025】

出力デバイス105は、例えばディスプレイやプリンタに代表される装置であって、本装置で処理した画像を表示やプリントアウト等、様々な形態で出力することのできる装置である。

【0026】

ネットワークインターフェース 106 は、他の装置が送信した画像や文字などの情報を、インターネットや LAN などのネットワークを介して本装置が受信するためのインターフェースである。

【0027】

107 は上記各部を繋ぐバスである。本実施形態に係る画像処理装置の基本構成は以下の説明上図 1 に示した構成とするが、これ以外の構成を適用することも可能であり、図 1 に示した構成に限定されるものではない。

【0028】

図 2 は、上記画像処理装置を含むシステムの概要を示す図である。201, 202 は夫々上記画像処理装置を適用するクライアント端末装置であって、後述するサーバ装置 204 とネットワーク 203 を介して各種のデータ通信を行うことができる。

【0029】

203 はネットワークで、インターネットや LAN などの有線のネットワーク、無線のネットワークを含むものである。

【0030】

204 はサーバ装置で、上述の通りネットワーク 203 を介してクライアント端末装置 201, 202 とデータ通信を行うことができ、例えばクライアント端末装置 201 やクライアント端末装置 202 から所望の画像の要求があった場合には、画像の符号化データを大量に保存する記憶装置 205 から要求された分だけのデータをネットワーク 203 を介して返信するものである。この記憶装置 205 は例えばハードディスクドライブ装置や、CD-ROM や DVD-ROM などの記憶媒体からプログラムやデータを読み取る装置がこれに該当する。

【0031】

本実施形態ではこの記憶装置 205 には、JPEG 2000 に従った符号化方式で符号化された画像のデータが複数保存されているものとする。従ってクライアント端末装置 201, 202 はサーバ装置 204 に、記憶装置 205 が保存している画像のデータの中から所望の画像について要求する。

【0032】

以下の説明では、サーバ装置 2 0 4 から受信した画像のデータをクライアント端末装置がキャッシュする方法について説明する。クライアント端末装置が記憶装置 2 0 5 に保存されている画像をダウンロードするためには、Web ブラウザを用いてサーバ装置 2 0 4 にアクセスする必要がある。サーバ装置 2 0 4 はこのアクセスを受け、記憶装置 2 0 5 に保存されている画像の一部もしくは全部の画像を例えばサムネイル形式でクライアント端末装置に提示する。これにより Web ブラウザにはこれらの画像が表示されることになる。

【 0 0 3 3 】

この Web ブラウザに表示された画像群の中から、クライアント端末装置のユーザが入力デバイス 1 0 4 を用いて所望の画像を指示すると、クライアント端末装置は条件（画面サイズや表示形態など）に応じた画像サイズや解像度に従って、所望の画像のデータ中の断片的なデータの送信要求をサーバ装置 2 0 4 に対して送信する。サーバ装置 2 0 4 はこの要求に応じた断片的なデータをクライアント端末装置に送信するので、クライアント端末装置はこれを受信し、バッファにキャッシュする。

【 0 0 3 4 】

そして受信した画像を表示する際には、それらのキャッシュデータから J P E G 2 0 0 0 のシンタックスに準拠したビットストリームを作成し、その後、そのビットストリームをデコードして表示する。

【 0 0 3 5 】

次に、一般的な J P E G 2 0 0 0 に従ったビットストリームについて説明する。図 3 は Layer-resolution level-component-position progression（以下、LRCP と記す）に従った J P E G 2 0 0 0 のビットストリームの構成を示す図である。J P E G 2 0 0 0 のビットストリームにはメインヘッダ（Main Header）、タイルヘッダ（Tile Header）、データ（Data）が含まれており、このデータの部分に画像符号化データが記録されることになる。またタイルヘッダは、圧縮符号化元の画像を所定のサイズの矩形（タイル）毎に分割し、夫々のタイルの圧縮符号化データを生成した場合に、夫々のタイルの圧縮符号化データに対して作成されるものである。メインヘッダ、タイルヘッダについては公知の技術によるもの

であるのでここでの説明は省略する。

【0036】

L R C P に準じた場合、画像の符号化データ（同図において” D a t a ” で示された部分）は、Layer/Resolution/Component/Positionの順にデータが配置された構成を備え、この構成はprogression orderと呼ばれる。また、ここで記されているpositionとは、JPEG2000符号化データにおけるprecinctのことである。

【0037】

J P E G 2 0 0 0 の符号化データは大まかにはLayer毎のデータに分けることができる。各Layerのデータはすなわち公知のビットプレーン符号化による各ビットプレーンの符号化データで、MSB側のビットプレーン（L a y e r 1）から順にLSBのビットプレーン（L a y e r L）までが配置される。Layer番号は復元する画像の原画に対するS/N比に対応し、Layer番号が小さいほどS/N比が悪く（低く）なる。すなわち、同図のJ P E G 2 0 0 0 のデータは、S/Nの悪い順に各L a y e r のデータが配置されていることになる。

【0038】

更に各Layerのデータは各Resolutionのデータにより構成されている。各Resolutionのデータは解像度（画像のサイズ）に応じたResolution番号に従った順序で配置されている。解像度とResolution番号との関係を図4に示す。最も小さい解像度の画像のResolution番号を0とし、Resolution番号が一つ増加するごとに画像の幅と高さが倍になっていく。各Layer内は、Resolution番号の小さい順にデータが格納されている。

【0039】

図3に戻って、各R e s o l u t i o n のデータは各C o m p o n e n t のデータにより構成されている。各C o m p o n e n t のデータは画像の各色データに対応しており、例えば画像がR G B の各データにより構成されている場合には、C o m p o n e n t 0 のデータはR成分のデータ、C o m p o n e n t 1 のデータはG成分のデータ、C o m p o n e n t 2 のデータはB成分のデータである、すなわち、C o m p o n e n t 数は画像の色空間の次元数に一致する。

【0040】

また各Component データには圧縮符号化元の画像における空間的な各位置のデータ (Position データ) が順番に記録されており、各Position データには、空間的な順番通りに番号 (ポジション番号) が付けられている (例えば画像の左上隅を 0 として画像の右方向に 1 つずつ番号が増加し、右端に達したら 1 つ下、且つ左端からまた画像右方向に番号が増加する)。

【0041】

一つの J P E G 2 0 0 0 ファイル内でのResolution番号とレイヤ番号、コンポーネント番号の最大値は、エンコーダによって予め設定されており、圧縮符号化元の画像はそのパラメータに従ってエンコードされており、その情報はメインヘッダに記録されている。また各パケット (packet) は、そのパケットに格納されているコードブロック (code-block) の情報を管理しているパケットヘッダ (packet header) 部と、各コードブロックの符号化データから構成されている。同図では 1 つのPosition データがパケットに相当する。この「パケット」なるものは、論理単位の一つである。

【0042】

このような構造を有する J P E G 2 0 0 0 ファイルがサーバ装置に保持されていれば、クライアント端末装置は必要な解像度等、要求する部分の画像のみをサーバ装置から受信することができる。本実施形態の場合には、JPIPによってデータの送受信を行うため、受信するデータの単位としては、J P E G 2 0 0 0 のタイル単位、或いは J P E G 2 0 0 0 のパケット (packet) 単位が考えられる。本実施形態では、クライアント端末装置がサーバ装置から受信するデータ単位としてパケット単位を想定する。

【0043】

このパケット単位でのリクエスト及びレスポンスの概念図を図 5 に示す。同図ではサーバ装置 502、クライアント端末装置 501 間のリクエスト、レスポンスについて示している。

【0044】

同図は、画像がタイル分割されており、各タイルの符号化データ (同図ではTi

le0～TileNまでの各データ)がJ P E G 2 0 0 0ビットストリーム504としてサーバ装置102に接続されている記憶装置503に格納されている状態で、クライアント端末装置501から所望のタイル、所望の解像度、所望のレイヤ、所望のコンポーネント、そして所望の空間的な位置の符号化データを要求する場合に、サーバ装置502とクライアント端末装置501との間で行われる通信を示すものである。

【0045】

ここで例えばクライアント端末装置501がタイル番号は「1」、解像度番号は「0」、レイヤ番号は「1」、コンポーネント番号は「0」、そしてポジション番号は「0」である符号化データをサーバ装置502に要求した場合、サーバ装置502は記憶装置503に保存しているJ P E G 2 0 0 0ビットストリーム504を解析して、要求に相当する部分(要求されたタイル番号、解像度番号、レイヤ番号、コンポーネント番号、ポジション番号に相当する部分)、すなわち、タイル番号1の packets 0 をレスポンスデータとして抜き出し、クライアント端末装置501に返信する。

【0046】

このように本実施形態ではサーバ装置とクライアント端末装置との間の画像データの送受信にはJ P I Pを用い、パケット単位で画像の符号化データをサーバ装置からクライアント端末装置に送信する。

【0047】

またその際、サーバ装置はJ P I Pにおけるprecinct data-bin IDの順番にパケットを返送するものとする。このprecinct data-binについて、図6(a)、(b)を用いて簡単に説明する。図6(a)、(b)はprecinct data-binを説明するための図である。

【0048】

precinct data-binとは図6(a)に示すように、Tile t n 中の p r e c i n c t p n の r e s o l u t i o n l e v e l r n 、 c o m p o n e n t 番号 c n を構成する全ての l a y e r のパケットを、layer番号が昇順になるように並べてつなげたデータの塊である。サーバ装置からクライアント端

末装置への(JPIPの)レスポンスデータは、このprecinct data-binを基にして作成され、レスポンスデータに記述されたprecinct data-bin ID ($PrID(tn, rn, cn, pn)$ と表す)で、夫々のデータを識別できる。 $PrID(tn, rn, cn, pn)$ は、次式により求めることができる。

【0049】

$$PrID(tn, rn, cn, pn) = tn + (cn + s \times (\text{component 数})) \times \text{tile 数}$$

ただし、

$$s = pn + tn \times (\text{resolution level } rn \text{ における } 1 \text{ tile あたりの precinct 数}) + (\text{resolution level } 0 \text{ から } (rn - 1) \text{ までのタイル } tr \text{ の precinct 数の総和})$$

従って例えば、2つのresolution levelを持ち、3つのcomponent、4つのタイル、そして、1タイルにつき1positionであるような画像の場合、 $PrID(tn, rn, cn, pn)$ は、

$$PrID(tn, rn, cn, pn) = tr + (c + 3s) \times 4$$

で求められ、各resolution level、各componentにおける各タイルのデータには図6(b)に示すような番号が付けられる。このようにして、precinct data-bin中の各resolution level、各componentにおける各タイルのデータには固有の番号であるPrIDが付けられるので、夫々のタイルのデータを識別することができる。

【0050】

また、記憶装置205に保持されている画像の符号化データは、以下のエンコード条件に従って作成されているものとする。

【0051】

最大解像度時の画像サイズ: 2048×2048[pixel]

Tile分割: 8×8=64 tileに分割

Resolution level数: 4 (i.e. resolution level 0~3)

Layer数: 3 (i.e. layer 0~2)

Position数：1 position / 1 tile

Component数：3 (i.e. component 0~2)

Progression order: Resolution-Layer-Component-Position

図7は、このようなエンコード条件で符号化された画像の符号化データ、即ち J P E G 2 0 0 0 のビットストリームの構成を示す図である。また同図において各タイルに付けられた番号は、図8に示すように、符号化元の画像の左上から順番に付けられたものである。

【0052】

次に、図2に示したクライアント端末装置201（202でも良い）が行う、サーバ装置204から所望の画像の符号化データを受信してキャッシュするまでの処理について以下説明する。

【0053】

図9は、後述するキャッシュファイルを作成する為の処理のフローチャートである。なお同図のフローチャートに従ったプログラムは二次記憶メモリ103に保存されており、CPU101による制御の下で一次記憶メモリ102にロードされ、CPU101がこれを実行することで、クライアント端末装置201（202でも良い）は以下説明する処理を行うことになる。

【0054】

まずクライアント端末装置201はサーバ装置204に、これから表示しようとする画像の符号化データを含むビットストリーム中のメインヘッダの送信を要求する（ステップS901）。

【0055】

受信したメインヘッダは一次記憶メモリ102に記憶され、CPU101はメインヘッダの受信を検知すると（ステップS902）、このメインヘッダを解析し、これから表示しようとする画像の基本情報、即ち、この画像の高さI_hと幅I_w、タイル数TN、resolution level数RN、layer数LN、component数CN、各resolution level rにおいて1タイルあたりのposition数PoN(r)、1タイルあたりに含まれるpacket数PaNを求める（ステップS903）。これらの値は、メインヘッダの中のSIZマーカ及びCODマーカを解析することで、容易に得られる。本実施形

態では画像の基本情報を、

$$I h = I w = 2048$$
$$T N = 64、$$
$$R N = 4$$
$$L N = 3$$
$$C N = 3$$
$$P o N (0) = P o N (1) = P o N (2) = P o N (3) = 1$$
$$P a N = 36$$

であると仮定するが、以下の説明の趣旨はこれらの値に限定されるものではない。

【0056】

次に、ステップS903で得られた画像の基本情報、これから受信する各パケットデータを管理するためのデータ、及びステップS902で受信したメインヘッダを書き込んだキャッシュファイルのデータを一次記憶メモリ102上に作成する（ステップS904）。

【0057】

図10は本実施形態に係るキャッシュファイルの概略構成を示す図である。1001はキャッシュファイル本体であり、上述の通り、キャッシュファイルには画像の基本情報、メインヘッダ、そしてこれから受信する各パケットに関する情報を登録し、管理するための情報（管理情報）が含まれる。タイル1を例に取り、このタイル1に含まれるパケットを管理するための情報1002の詳細を同図の右側に示す。

【0058】

タイル1に含まれるパケットを管理するための情報として、夫々のパケットについて以下の4つの情報を管理する。

【0059】

- ・ キャッシュファイルの先頭から、このキャッシュファイルに以降付加（キャッシュ）されていく各パケットのデータまでのオフセット値（Offset）
- ・ パケットのデータ長（Length）

- ・ タイル内での各パケットの順番を示す番号packet sequence number (SeqNumber)
- ・ 各パケットのデータの連続性を示すフラグ (SeqFlag)

即ち情報1002内には、タイル1に含まれる各パケットについて上記4つの情報のセットが登録されることになる。上記4つの各情報の詳細については後述する。

【0060】

また上述のOffset、Length、SeqFlagについては、ステップS904における作成時には、Offset=0、Length=0、SeqFlag=NON_IN_SEQUENCEと初期化されており、SeqNumberについてはPAN=36であるため、先頭から通し番号0～35が付けられる。この初期化された状態の管理情報を図11(a)に示す。

【0061】

以上の処理により、管理情報を初期化することができ、この初期化された管理情報に以降受信する各パケットに関する上記4つの情報を管理情報に登録することができる。

【0062】

次にクライアント端末装置201がサーバ装置204に画像の符号化データを要求し、要求に応じてサーバ装置204がパケット毎に送信する画像の符号化データを受信し、受信した画像の符号化データをデコードして表示装置としての出力デバイス105に表示するまでの処理の詳細について、同処理のフローチャートを示す図12を参照して以下説明する。なお同図のフローチャートに従ったプログラムは二次記憶メモリ103に保存されており、CPU101による制御の下で一次記憶メモリ102にロードされ、CPU101がこれを実行することで、クライアント端末装置201は以下説明する処理を行うことになる。

【0063】

まず、入力デバイス104を用いてユーザから指示された画像の表示要求を取得する(ステップS1201)。CPU101はこの要求に基づいて画像全体をどのような解像度で表示するのかといったことを示すデータ(要求データ)を作

成し、サーバ装置 204 に送信する（ステップ S1202）。本実施形態では画像全体を resolution level 1、layer 0 で表示するので、要求データはサーバ装置 204 に対して resolution level 0~1、layer 0、component 0~2 のパケットデータを送信するよう、要求するためのものである。

【0064】

従ってサーバ装置 204 はこの要求を受けて、resolution level 0~1、layer 0、component 0~2 のパケットデータを全タイルに関してクライアント端末装置 201 に送信することになるので、クライアント端末装置 201 はサーバ装置 204 からのパケット単位でのデータ（レスポンスデータ）を受信すると（ステップ S1203）、後述する方法でキャッシュする（ステップ S1204）。受信した各パケットのデータは一次記憶メモリ 102 の受信バッファに記憶されるものとする。なおステップ S1204 に関する処理についての詳細は後述する。

【0065】

そして要求した全てのパケットのデータを受信すると（ステップ S1205）、キャッシュファイルから JP2 ビットストリームを作成し（ステップ S1206）、JP2 ビットストリームをデコードし、出力デバイス 105 に表示する（ステップ S1207）。

【0066】

ここで上記ステップ S1204 における処理、即ちサーバ装置 204 からパケット単位で受信した画像の符号化データをキャッシュする処理の詳細について、図 13 を参照して説明する。図 13 は、サーバ装置 204 からパケット単位で受信した画像の符号化データをキャッシュする処理の詳細のフローチャートである。

【0067】

まず、受信すべき総パケット数 Resp_number を取得する（ステップ S1301）。取得の方法としては例えば上記要求データが示す内容から求めても良いし（resolution level 0~1、layer 0、component 0~2 のパケットの数を求める）、実際に受信したパケットの数をカウントしても良い。

【0068】

なお本実施形態では1タイルでresolution level 0~1、layer 0、component 0~2を構成するパケットの数、すなわち6個のパケットがサーバ装置204から送信されるので、受信すべき総パケット数は6（1タイルあたりのパケット数）×64（総タイル数）=384となる。

【0069】

次に、サーバ装置204から送信される夫々のパケットに関する情報を一時的に格納するための配列Infoを作成する（ステップS1302）。配列Infoに格納するための「パケットに関する情報」は以下の4つの項目から成る。

【0070】

- ・ パケットが属するタイルの番号（T i n d e x）
- ・ 属するタイル内での順番を表す番号Packet sequence number（s e q N u m b e r）
- ・ 上記受信バッファの先頭から、パケットのデータまでのオフセット値（B o f f s e t）
- ・ パケットのデータ長（L e n g t h）

図14は配列Infoの構成を示す図である。同図に示すとおり、配列Infoは、上記4つの情報を1セットとし、このセットを各パケットについて格納するための配列である。上述の通り、送信すべき総パケット数は384であるので、配列Infoは384個設けられている。

【0071】

同図においてInfo[0]は最も小さいタイル番号（本実施形態では0）のタイルにおいて順番が最初のパケットに関する上記4つの情報を格納するための領域で、1401、1402、1403、1404は夫々同図に示すとおり、最も小さいタイル番号のタイルにおいて順番が最初のパケットに対するT i n d e x、s e q N u m b e r、B o f f s e t、L e n g t hを格納するための領域である。

【0072】

各パケットについてのこれら4つの情報は、上記受信バッファに記憶している各パケットのデータを解析したり、上記P r I Dを求める式を用いたりして求め

ることができる。

【0073】

よって、クライアント端末装置201はパケットの受信順に従って、パケットに関する上記4つの情報を配列Infoに格納する（ステップS1303）。

【0074】

本実施形態ではサーバ装置204は、precinct data-bin順にレスポンスデータを送信するので、配列InfoにおけるTindex、seqNumberは以下のような並びになる。

【0075】

Info[0].Tindex=0, Info[0].seqNumber=0

Info[1].Tindex=1, Info[1].seqNumber=0

Info[2].Tindex=2, Info[2].seqNumber=0

...

Info[64].Tindex=0, Info[64].seqNumber=1

Info[65].Tindex=1, Info[65].seqNumber=1

Info[66].Tindex=2, Info[66].seqNumber=1

...

Info[128].Tindex=0, Info[128].seqNumber=2

Info[129].Tindex=1, Info[129].seqNumber=2

Info[130].Tindex=2, Info[130].seqNumber=2

...

次に、配列Infoをタイル番号順にソートする（ステップS1304）。すなわち、配列Info[x]（x=0, 1, 2, , , 383）のTindexを参照し、値が小さい順に配列Info[x]を並び替える。ソートの方法としては、クイックソートなどの方法が考えられるが、その方法については特に限定するものではない。

【0076】

本実施形態ではステップS1304における並び替えの処理によって、配列Infoの並びが以下になったとする。

【0077】

Info[0].Tindex=0, Info[0].seqNumber=0
Info[1].Tindex=0, Info[1].seqNumber=1
Info[2].Tindex=0, Info[2].seqNumber=2
Info[3].Tindex=0, Info[3].seqNumber=9
Info[4].Tindex=0, Info[4].seqNumber=10
Info[5].Tindex=0, Info[5].seqNumber=11
Info[6].Tindex=1, Info[6].seqNumber=0
Info[7].Tindex=1, Info[7].seqNumber=1

...

次に、タイル番号が同じ配列 *I n f o* 同士を、*seqNumber* が示す番号順にソートする（ステップ S 1 3 0 5）。例えば配列 *I n f o* の上記並びにおいて *T i n d e x* が 0 である配列は先頭から 6 つあるので、この 6 つの配列を *s e q N u m b e r* が示す値が小さい順に並び替える。同様の処理を *T i n d e x* が 1 以降についても行う。この場合のソートの方法としても、クイックソートなどの方法が考えられるが、その方法については特に限定するものではない。

【0078】

ステップ S 1 3 0 4、ステップ S 1 3 0 5 による処理の結果、配列 *I n f o* は各タイル内で *packet sequence number* 順に並んでいたもので、このステップ S 1 3 0 4、ステップ S 1 3 0 5 による処理の前後で配列 *I n f o* の並びは上記並びと同じである。

【0079】

なお図示していないがステップ S 1 3 0 4、ステップ S 1 3 0 5 では、受信バッファに記憶されている各パケットのデータも同様に並び替えるものとする。即ち、受信バッファに記憶されている各パケットのデータを、含まれるタイルの番号順に並び替え、更に、含まれるタイル内の順番に従って並び替える。

【0080】

次に、並び替えられた各パケットのデータを、並び替えた順番で上記キャッシュファイルに付加すると共に、各パケットを管理するための情報としての上記管

理情報に上記付加された夫々のパケットに関する情報を登録する（ステップ S 1 3 0 6）。

【0081】

次に、上記ステップ S 1 3 0 6 における処理の詳細を図 1 5 に示すフローチャートを参照して以下説明する。図 1 5 は、ステップ S 1 3 0 6 における処理の詳細を示すフローチャートである。

【0082】

まず受信した各パケットをカウントする為の変数 I に 1 を代入して初期化する（ステップ S 1 5 0 1）。次に、配列 I n f o において、同じ T i n d e x の値を有し、且つ s e q N u m b e r が連続する数をカウントする為の変数 X に 1 を代入して初期化する（ステップ S 1 5 0 2）。

【0083】

次に、（I - 1）番目の配列 I n f o のタイル番号 T i n d e x と I 番目の配列 I n f o のタイル番号 T i n d e x とを比較し、夫々が同じ値であるか否か、換言すれば、（I - 1）番目の配列 I n f o と I 番目の配列 I n f o とが、同じタイルに含まれるパケットに関する情報を格納しているのかを否かを判断する（ステップ S 1 5 0 3）。

【0084】

（I - 1）番目の配列 I n f o のタイル番号 T i n d e x と I 番目の配列 I n f o のタイル番号 T i n d e x とが同じ値、即ち、（I - 1）番目の配列 I n f o と I 番目の配列 I n f o とが、同じタイルに含まれるパケットに関する情報を格納している場合には処理をステップ S 1 5 0 4 に進め、（I - 1）番目の配列 I n f o の s e q N u m b e r が示す値に 1 を加えた値と I 番目の配列 I n f o の s e q N u m b e r が示す値とを比較し、夫々が同じ値であるか否か、換言すれば、（I - 1）番目の配列 I n f o と I 番目の配列 I n f o とが、順番が連続している（順番が隣接している）パケットに関する情報を格納しているのかを否かを判断する（ステップ S 1 5 0 4）。

【0085】

（I - 1）番目の配列 I n f o の s e q N u m b e r が示す値に 1 を加えた値

と I 番目の配列 `Info` の `seqNumber` が示す値とが同じ値、即ち、 $(I-1)$ 番目の配列 `Info` と I 番目の配列 `Info` とが、順番が連続している（順番が隣接している）パケットに関する情報を格納している場合には処理をステップ S1505 に進め、上記変数 X、I 夫々に 1 を加えて更新する（ステップ S1505）。

【0086】

一方、ステップ S1503 における判断の結果、 $(I-1)$ 番目の配列 `Info` のタイル番号 `TileIndex` と I 番目の配列 `Info` のタイル番号 `TileIndex` とが異なる値、即ち、 $(I-1)$ 番目の配列 `Info` と I 番目の配列 `Info` とが、夫々異なるタイルに含まれるパケットに関する情報を格納している場合、もしくは、ステップ S1504 における判断の結果、 $(I-1)$ 番目の配列 `Info` の `seqNumber` が示す値に 1 を加えた値と I 番目の配列 `Info` の `seqNumber` が示す値とが異なる値、即ち、 $(I-1)$ 番目の配列 `Info` と I 番目の配列 `Info` とが、順番が連続していない（順番が隣接していない）パケットに関する情報を格納している場合には処理をステップ S1508 に進め、後述するキャッシュファイルの更新処理を行う（ステップ S1508）。ステップ S1508 における処理の詳細については後述する。

【0087】

そしてキャッシュファイルの更新処理の後、上記変数 X に 1 を代入して初期化し、更に変数 I に 1 を加えて更新する（ステップ S1509）。そしてステップ S1505、ステップ S1509 の何れの処理の後にも変数 I が示す値が受信すべき総パケット数 `Resp_number` と同じ値であるか否かを判断する（ステップ S1506）。同じでない場合には処理をステップ S1503 に戻し、以降の処理を行う。

【0088】

一方、同じでない場合には処理をステップ S1507 に進め、ステップ S1508 と同じキャッシュファイルの更新処理を行う。ステップ S1507 における処理はステップ S1508 における処理と同じであるので、以下、まとめてこれらの処理について説明する。

【0089】

図16はステップS1507, ステップS1508におけるキャッシュファイルの更新処理の詳細を示すフローチャートである。キャッシュファイルの更新処理とは上述の通り、同じタイルに含まれ、且つ同じタイル内で順番が連続したパケット群のデータをキャッシュファイルに順次付加（キャッシュ）し、更に、このパケット群に関する情報をキャッシュファイルに含まれる管理情報に登録する処理である。管理情報に対して登録する情報は、配列 `Info[I-X]` から `Info[I-1]` に格納されている。

【0090】

なお、本実施形態の場合、配列 `Info` において最初にこのキャッシュファイルの更新処理の対象となるのは、配列 `Info[0]`, `Info[1]`, `Info[2]` であって、夫々の配列における `Tindex`, `seqNumber`, `Length` は以下の通りとなる。

【0091】

`Info[0].Tindex=0, Info[0].seqNumber=0, Info[0].Length=64`

`Info[1].Tindex=0, Info[1].seqNumber=1, Info[1].Length=72`

`Info[2].Tindex=0, Info[2].seqNumber=2, Info[2].Length=38`

また、これらの配列に対するキャッシュファイルの更新処理を行う時点での変数 `X`, `I` の値は共に3である。

【0092】

キャッシュファイルの更新処理において、まずキャッシュファイルのファイルサイズ `Fsize` を取得する（ステップS1601）。本実施形態では、`Fsize=38630 [byte]` と仮定する。

【0093】

次に、配列 `Info` の番号をカウントする変数 `an` に $(I-X)$ が示す値を代入し、更に、キャッシュファイルの先頭から、このキャッシュファイルに以降付加（キャッシュ）されていく各パケットのデータの先頭までのオフセット値を代入するための変数 `sp` にステップS1601で取得したファイルサイズ `Fsize` が示す値を代入し、初期化する（ステップS1602）。

【0094】

本実施形態ではこの時点で

$a_n = 0$

$sp = 38630$

となっている。

【0095】

そして以下説明するステップS1603からステップS1608までの処理を対象とする全ての packets (上記並び替えられた packets のうち、(I-X) 番目から (I-1) 番目の packets) に対して行うことで、対象とする全ての packets に関する情報を管理情報に登録することができると共に、対象とする全ての packets のデータをキャッシュファイルに順次付加 (キャッシュ) することができる。以下、ステップS1603からステップS1608までの各処理について説明する。

【0096】

先ず管理情報において、 $Info[a_n].Tindex$ をタイル番号とするタイルにおいて、 $Info[a_n].seqNumber$ で表される順番の packets (以下、packets a_n と呼称する) を管理するための情報 ($Offset$, $Length$, $seqNumber$, $seqFlag$) において、 $Offset$ に上記変数 sp が示す値を代入し、 $Length$ に $Info[a_n].Length$ が示す値を代入する (ステップS1603)。

【0097】

本実施形態では最初にタイル番号が0, $seqNumber$ が0の packets を管理するための情報が更新され、この packets に関する情報のうち $Offset$ に上記変数 sp が示す値38630が代入され、 $Length$ に $Info[0].Length$ が示す値64が代入される。

【0098】

これにより、キャッシュファイルの先頭から38630バイト目以降、64バイト分のところにタイル番号が0, $seqNumber$ が0の packets のデータが付加 (キャッシュ) されるという旨、即ちタイル番号が0, $seqNumber$

r が 0 のパケットのデータのキャッシュファイルにおける配置情報を登録することができる。

【0099】

次に、管理情報におけるパケット a n の S e q F l a g を更新する。いうまでもなく、配列 I n f o [I - X] から配列 I n f o [I - 1] までの各配列には連続した順番のパケットに関する情報が格納されている。従って、a n が (I - 1) でない限り、管理情報においてパケット a n を管理するための情報の次には連続してパケット (本実施形態の場合タイル番号が 0, s e q N u m b e r が 1 のパケット) を管理するための情報が登録されることになるので、パケット a n の S e q F l a g にはこのことを示す値 IN_SEQUENCE (パケット a n の次に連続してパケットを管理するための情報が登録されない場合には NON_IN_SEQUENCE) を代入する。

【0100】

このことから、図 16 に示すフローチャートでは、変数 a n が示す値が (I - 1) が示す値と同じであるか、即ち、処理が対象とする最後のパケットまで達しているのかを判断する (ステップ S 1604)。達していない場合 (本実施形態では a n = 0, 1 の場合) には処理をステップ S 1605 に進め、達している場合 (本実施形態では a n = 2 の場合) にはステップ S 1605 をスキップして処理をステップ S 1606 に進める。

【0101】

そして上述の通り、達していない場合 (本実施形態では a n = 0, 1 の場合) にはパケット a n の S e q F l a g には値 IN_SEQUENCE を代入する (ステップ S 1605)。本実施形態ではタイル番号が 0、s e q N u m b e r が 0, 1 であるパケットについては S e q F l a g には値 IN_SEQUENCE が代入される。

【0102】

次に、変数 s p にパケット a n の長さを示す変数 I n f o [a n] . L e n g t h を加え、キャッシュファイルの先頭からパケット (a n + 1) の先頭までのオフセット値を求めて再度変数 s p に代入する (ステップ S 1606)。本実施形態では a n = 0 の時には s p = 38630 + I n f o [0] . L e n g t h =

$38630 + 64 = 38694$ となり、これがキャッシュファイルの先頭からタイル番号が 0, `seqNumber` が 1 のパケット (上記パケット ($a_n + 1$) に相当) のデータの先頭までのオフセット値となる。

【0103】

そして次のパケットであるパケット ($a_n + 1$) に関する情報を管理情報に登録する対象とするために、変数 a_n に 1 を加える (ステップ S1607)。そして次に、変数 a_n が示す値が変数 I が示す値と同じであるか、即ち、配列 `Info` [$I - X$] から配列 `Info` [$I - 1$] までの各配列に格納されたパケットに関する情報を管理情報に登録したかを判断し (ステップ S1608)、全て登録していない場合には処理をステップ S1603 に戻し、以降の処理を繰り返す。

【0104】

一方、全て登録した場合には処理をステップ S1609 に進め、受信バッファに記憶された、並び替えられた各パケットのデータのうち、($I - X$) 番目から X 個分 (($I - 1$) 番目まで) の各パケットのデータを順番にキャッシュファイルに付加 (キャッシュ) する (ステップ S1609)。

【0105】

受信バッファに記憶された、並び替えられた各パケットのデータのうち、キャッシュファイルに付加 (キャッシュ) するパケットのデータの特定については、上記受信バッファの先頭から、`Info` [z]. `Boffset` バイト目のデータから `Info` [z]. `Length` バイト分のデータを抜き出し、キャッシュファイルに付加する (ただし、 z は ($I - X$) から ($I - 1$) までの値である)。

【0106】

以上の処理により、図 11 (a) に示した管理情報は、図 11 (b) に示した管理情報に更新される。同図に示すように、キャッシュファイルの最後に付加した各パケットのデータのキャッシュファイルにおける配置については管理情報に `Offset`、`Length` として登録されている。本実施形態の場合、0 番目から 2 番目までのパケットのデータ、即ち、タイル番号が 0 で、`packet sequence number` が 0, 1, 2 の各パケットのデータがこの順番でキャッシュファイルに

付加されることになる。

【0107】

以上の説明により、本実施形態によって、受信した各パケットのデータを連続して、順番にキャッシュすることができるので、これらのパケットを読み出す際にはまとめて読み出すことができる。これにより読み出しのためのキャッシュファイルへのアクセス数が軽減される。

【0108】

また、読み出す際に、キャッシュファイル中の各パケットのデータの配置が管理情報に登録されているので、キャッシュファイル中における各パケットのデータの位置を特定することが容易となり、各パケットのデータを読み出す処理をより高速に行うことができる。

【0109】

[第2の実施形態]

本実施形態では、第1の実施形態で生成したキャッシュファイルから、J P E G 2 0 0 0 のビットストリームを生成する処理について説明する。なお、以下ではキャッシュファイル中のメインヘッダは第1の実施形態で取得したメインヘッダ、そしてパケットのデータはresolution level 1、layer 0、3 componentを構成するパケットのデータとするが、以下説明する趣旨はこれに限定されるものではない。ただし、未受信のパケットのデータの部分には、zero length packetをダミーデータとして利用する。

【0110】

図17は、キャッシュファイルから、各タイルのパケットのデータを出力ファイルに読み出す処理のフローチャートである。なお同図のフローチャートに従ったプログラムは二次記憶メモリ103に保存されており、CPU101による制御の下で一次記憶メモリ102にロードされ、CPU101がこれを実行することで、クライアント端末装置201（202でも良い）は以下説明する処理を行うことになる。

【0111】

まずキャッシュファイルから画像の基本情報を取り出し、タイル数TNを取得

する（ステップS1701）。本実施形態ではキャッシュファイルは第1の実施形態で作成されたものを用いるので、画像の基本情報も当然第1の実施形態で説明したものと同一であるので、 $TN=64$ である。

【0112】

次に、キャッシュファイルからメインヘッダデータを読み出し、出力ファイルに出力する（ステップS1702）。なお、出力ファイルは一次記憶メモリ102上に作成されるものとする。この出力ファイルには後述の処理によってタイル毎にパケットのデータを出力するので、夫々のタイルの番号をカウントする為の変数が必要となる。よってこの変数を t とし、この変数 t を0に初期化する（ステップS1703）。

【0113】

次に、タイル番号 t のタイルヘッダを取得し、出力ファイルに書き込む（ステップS1704）。本実施形態では、全てのタイルが同じエンコード条件のため、タイルの管理情報から、タイル番号、キャッシュから作成されるタイルのデータ長を取得することで、タイルヘッダを簡単に作成・取得することができる。

【0114】

次に、画像の基本情報を参照し、タイル番号 t のタイルに含まれるパケットの総数 PaN を取得する（ステップS1705）。画像の基本情報は当然第1の実施形態で説明したものと同一であるので $PaN=36$ となる。そしてキャッシュファイルから読み出すパケットのデータをカウントする為の変数 k を0に初期化する（ステップS1706）。

【0115】

そして次に、キャッシュファイル中の各パケットを管理するための情報（管理情報）から、タイル番号 t のタイル中のパケット群で上記変数 k が示す値と同じ値である $seqNumber$ に対応するパケットの情報（ $Offset$, $Length$, $SeqNumber$, $SeqFlag$ ）をキャッシュファイルから取得する（ステップS1707）。

【0116】

まず各変数 t 、 k は共に0に初期化されているので、本実施形態では次のデー

タを取得することになる（図11（b）に示した管理情報において、最初のパケットの情報、即ち、タイル番号0のタイル中のseqNumberが0であるパケットの情報）。

【0117】

Offset = 38630
Length = 64
SeqNumber = 0
SeqFlag = IN_SEQUENCE

次に以下説明する各変数を初期化する（ステップS1708）。キャッシュファイルに付加した各パケットのデータにおいて、同じタイルに含まれ、且つ同じタイル内で順番が連続したパケットの数をカウントする為の変数yに1を代入して初期化する。

【0118】

また、キャッシュファイルの先頭から読み出すパケットのデータの先頭までのオフセット値、即ち、これから読み出そうとするパケットのデータのキャッシュファイルにおける読み出し位置を保持するための変数ReadPointに、ステップS1707で取得したOffsetが示す値を代入する。

【0119】

また、読み出すパケット群（seqNumberの値が連続する値であるパケット群）のデータ長を保持するための変数ReadByteに、ステップS1707で取得したLengthが示す値を代入する。本実施形態では $t=0$ 、 $k=0$ の場合には、これら3つの変数が示す夫々の値は以下のようになる。

【0120】

y = 1
ReadPoint = 38630
ReadByte = 64

次に、ステップS1707で得られたseqFlagが示す値が、IN_SEQUENCEであるか（換言すれば、タイル番号tのタイル中のseqNumberがkであるパケットのデータとタイル番号tのタイル中のseqNumberが $(k+$

1) であるパケットのデータとが連続してキャッシュファイルに付加されているか) を判断する (ステップ S1709)。

【0121】

seqFlag が IN_SEQUENCE である場合には、処理をステップ S1710 に進め、そうでない場合には処理をステップ S1713 に進める。例えば $t = 0$, $k = 0$ の場合、本実施形態では図 11 に示すとおり seqFlag は IN_SEQUENCE であるので、処理はステップ S1710 に進めることになる。そして上記変数 k 、 y が保持する値を一つインクリメントし (ステップ S1710)、次のパケット (タイル番号が 0 のタイル中のパケット群でインクリメントされた変数 k ($= 1$) が示す値と同じ値である seqNumber に対応するパケット) の情報 (Offset, Length, SeqNumber, SeqFlag) をキャッシュファイルから取得する (ステップ S1711)。

【0122】

変数 $t = 0$ 、 $k = 1$ であるので、本実施形態では次のデータを取得することになる (図 11 (b) に示した管理情報において、次のパケットの情報、即ち、タイル番号 0 のタイル中の seqNumber が 1 であるパケットの情報)。

【0123】

Offset	=	38694
Length	=	72
SeqNumber	=	1
SeqFlag	=	IN_SEQUENCE

そして次に、読み出すパケット群 (seqNumber の値が連続する値であるパケット群) のデータ長を保持するための変数 ReadByte に、ステップ S1711 で取得した情報に含まれる Length が示す値を加算する (ステップ S1712)。ステップ S1710 からステップ S1712 までの処理をステップ S1711 で取得した情報における seqFlag が IN_SEQUENCE である限り、即ち、seqNumber が連続した数値である限り繰り返して行う。

【0124】

この繰り返し処理により、変数 ReadByte には、タイル番号が t で、読み出すパ

ケット群 (seqNumberの値が連続する値であるパケット群) のデータ長が格納されることになる。

【0125】

本実施形態では $t = 0$, $k = 2$ になった場合に、処理をステップ S1713に進める。本実施形態でステップ S1713に進んだ時における各変数が示す値は以下のようにになっている。

【0126】

```
t = 0,    k = 2
Offset    = 38768
SeqNumber = 2
SeqFlag   = NON__IN__SEQUENCE
y         = 3
ReadPoint = 38768
ReadByte  = 174
```

次に、変数ReadPointの値が0であるか否かを判断する (ステップ S1713)。ReadPointの値が0ではない場合には処理をステップ S1714に進め、キャッシュファイルの先頭からReadPointバイト目からReadByteバイト分のデータを読み出し、出力ファイルに書き込む (ステップ S1714)。本実施形態の場合、キャッシュファイルの先頭から38768バイト目から174バイト分のデータ、即ち、タイル番号が0でseqNumberが0, 1, 2のパケットのデータをキャッシュファイルから読み出し、出力ファイルに書き込む。

【0127】

一方、ReadPointの値が0である場合には処理をステップ S1715に進め、パケットのデータはないと判断し、zero length packetデータを出力ファイルに書き込む (ステップ S1715)。そして次に読み出すパケットをカウントする為の変数kの値を一つインクリメントし (ステップ S1716)、インクリメント後の変数kの値がタイル番号tのタイルに含まれるパケットの総数PaNに達したか否かを判断する (ステップ S1717)。

【0128】

$k = P a N$ の場合、タイル番号 t のタイルに含まれるパケットのデータは全て出力ファイルに出力したものと判断し、次のタイルを処理対象とする為に、タイル番号を表す変数 t の値を一つインクリメントする（ステップ S 1 7 1 8）。本実施形態では $k = P a N = 36$ となった場合に、タイル番号 0 のタイル中の 36 個のパケットのデータが出力ファイルに書き込まれたものと判断する。

【0129】

一方、 $k < P a N$ の場合、タイル番号 t のタイルに含まれるパケットのデータは全て出力ファイルに出力されていないと判断し、処理をステップ S 1 7 0 6 に戻し、以降の処理を行う。

【0130】

そして $k = P a N$ の場合、インクリメント後の変数 t の値がタイル数 $T N$ に達したか否かを判断し（ステップ S 1 7 1 9）、 $t = T N$ の場合、全てのタイルについてパケットのデータを出力ファイルに書き込んだものと判断し、本処理を終了する。一方、 $t < T N$ の場合、全てのタイルについてパケットのデータを出力ファイルに書き込んでいないものと判断し、次にタイルについてパケットのデータの出力ファイルへの書き込み処理を行うために、処理をステップ S 1 7 0 4 に戻し、以降の処理を行う。

【0131】

以上の処理によって出力ファイル、換言すれば J P E G 2 0 0 0 のビットストリームが生成されることになる。

【0132】

以上の説明により、本実施形態によれば、連続してキャッシュされているパケットのデータを 1 回のファイルリードで取得することができる。これによりファイルのシーク回数を減らすことができ、キャッシュファイルから J P E G 2 0 0 0 のビットストリームを生成する処理がより高速化されることになる。

【0133】

[第3の実施形態]

本実施形態では、上記 $s e q F l a g$ を用いずに、第 1, 2 の実施形態と同様の目的を達成する画像処理について説明する。

【0134】

まずキャッシュファイルの作成処理について説明すると、本実施形態では seq F l a g は用いない。従って本実施形態に係るキャッシュファイルの作成処理のフローチャートは、図9に示したフローチャートのステップS904において、各パケットについて seq F l a g を除く他の3つについてのみ管理情報に含めるように変更したものとなる。またこれに伴って、本実施形態に係るキャッシュファイルの更新処理のフローチャートは図18に示すように、図16に示したフローチャートにおいて、seq F l a g に係る処理即ちステップS1604、S1605を省いたものとなり、その他については同じである。

【0135】

従ってこのような処理によって作成されるキャッシュファイルの構成は図19(b)に示すようになる。図19(b)は本実施形態に係るキャッシュファイルの作成処理によって作成されたキャッシュファイルの構成を示す図で、図19(a)は、第1の実施形態に係るキャッシュファイルの作成処理によって作成されたキャッシュファイルの構成を示す図である。双方を比較しても分かるように、例えばあるタイルの1番目のパケットの管理情報を比較しても、管理情報1003にはseq F l a g が含まれているのに対して、管理情報1902にはseq F l a g が含まれておらず、その他については管理情報1003と同じである。

【0136】

次に、本実施形態に係るキャッシュファイルから、J P E G 2 0 0 0 のビットストリームを生成する処理について説明する。図20は、キャッシュファイルから、各タイルのパケットのデータを出力ファイルに読み出す処理のフローチャートである。なお同図のフローチャートに従ったプログラムは二次記憶メモリ103に保存されており、CPU101による制御の下で一次記憶メモリ102にロードされ、CPU101がこれを実行することで、クライアント端末装置201(202でも良い)は以下説明する処理を行うことになる。なお図20において、図17に示した各ステップと同じものについては同じステップ番号を付けており、その説明は既に説明済みであることから、ここではその説明は省略する。

ステップ S 1 7 0 1 からステップ S 1 7 0 7 までの処理では第 2 の実施形態と同様の処理を行う。次に以下説明する各変数を初期化する（ステップ S 2 0 0 1）。

【0137】

キャッシュファイルに付加した各パケットのデータにおいて、同じタイルに含まれ、且つ同じタイル内で順番が連続しているかを判定する為の変数 J にステップ S 1 7 0 7 で取得した O f f s e t の値を代入して初期化する。

【0138】

また、キャッシュファイルに付加した各パケットのデータにおいて、同じタイルに含まれ、且つ同じタイル内で順番が連続したパケットの数をカウントする為の変数 y に 0 を代入して初期化する。

【0139】

また、キャッシュファイルの先頭から読み出すパケットのデータの先頭までのオフセット値、即ち、これから読み出そうとするパケットのデータのキャッシュファイルにおける読み出し位置を保持するための変数 ReadPoint に、ステップ S 1 7 0 7 で取得した O f f s e t が示す値を代入する。

【0140】

また、読み出すパケット群（s e q N u m b e r の値が連続する値であるパケット群）のデータ長を保持するための変数 ReadByte に 0 を代入して初期化する。

【0141】

次に、上記初期化処理が終了すると、連続パケットカウント用変数 y の値とパケットカウント用変数 k の値を夫々一つインクリメントし、更に、連続性を判定する為の変数 j が有する値と、読み出すパケット群（s e q N u m b e r の値が連続する値であるパケット群）のデータ長を保持するための変数 ReadByte が有する値とに夫々 Length の値を足す（ステップ S 2 0 0 2）。

【0142】

変数 j の値に L e n g t h の値を足すことで変数 j には、キャッシュファイルの先頭から、タイル番号 t のタイル中のパケット群で変数 k が示す値と同じ値である s e q N u m b e r に対応するパケット（これをパケット k と呼称する）の

すぐ後にキャッシュファイルに付加されたパケット（これをパケット（ $k+1$ ）と呼称する）の先頭までのオフセット値が格納されていることになる。

【0143】

次に、変数 k の値がタイル番号 t のタイルに含まれるパケットの総数 $P a N$ に達したか否かを判断する（ステップS2003）。

【0144】

$k = P a N$ の場合、タイル番号 t のタイルに含まれるパケットのデータは全て出力ファイルに出力したものと判断し、処理をステップS1713に進める。

【0145】

一方、 $k < P a N$ の場合、タイル番号 t のタイルに含まれるパケットのデータは全て出力ファイルに出力されていないと判断し、処理をステップS2004に進め、パケット（ $k+1$ ）の管理情報を取得する（ステップS2004）。即ち、ステップS1707で取得したパケット（パケット k ）の管理情報の次のパケット（パケット（ $k+1$ ））の管理情報を取得する。

【0146】

そしてここで変数 J の値が $O f f s e t$ の値と同じであるかを判断する（ステップS2005）。 $J = O f f s e t$ の場合、これは即ち、パケット k とパケット（ $k+1$ ）とが連続してキャッシュファイルに付加されていることを意味する。従って、処理をステップS2002に戻し、更に連続して付加されているパケットのデータが存在するかをチェックする。

【0147】

一方、 $J \neq O f f s e t$ の場合、これは即ち、パケット k とパケット（ $k+1$ ）とが連続してキャッシュファイルに付加されていないことを意味する。従って、処理をステップS1713に進める。

【0148】

なお、本実施形態ではステップS1713以降の処理は図17と比べてステップS1716が存在しないだけでその他については図17と同じである。

【0149】

このように本実施形態によれば $s e q F l a g$ を用いないので、上記実施形態

と比べてもキャッシュファイルのサイズを小さくすることができる。この seq Flag は上記第 1, 2 の実施形態では各パケット毎に存在していたので、パケットの数が多ければそれだけの数必要となっていた。しかし本実施形態では seq Flag は用いないので、パケットの数が多ければ多いほど、その効果はより大きなものとなる。

【0150】

[その他の実施形態]

本発明の目的は、前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記録媒体（または記憶媒体）を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ（または CPU や MPU）が記録媒体に格納されたプログラムコードを読み出し実行することによっても、達成されることは言うまでもない。この場合、記録媒体から読み出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記録した記録媒体は本発明を構成することになる。

【0151】

また、コンピュータが読み出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているオペレーティングシステム（OS）などが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【0152】

さらに、記録媒体から読み出されたプログラムコードが、コンピュータに挿入された機能拡張カードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込まれた後、そのプログラムコードの指示に基づき、その機能拡張カードや機能拡張ユニットに備わる CPU などが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【0153】

本発明を上記記録媒体に適用する場合、その記録媒体には、先に説明したフロ

ーチャートに対応するプログラムコードが格納されることになる。

【0154】

【発明の効果】

以上の説明により、本発明によって、符号化データを受信してキャッシュする場合に、キャッシュしたデータに対してアクセスする際に要する時間や回数を軽減させることができる。

【図面の簡単な説明】

【図1】

PC（パーソナルコンピュータ）やワークステーションなどにより構成される、本発明の第1の実施形態に係る画像処理装置の基本構成を示すブロック図である。

【図2】

図1に示した画像処理装置を含むシステムの概要を示す図である。

【図3】

Layer-resolution level-component-position progression（以下、LRCPと記す）に従ったJPEG2000のビットストリームの構成を示す図である。

【図4】

解像度とResolution番号との関係を示す図である。

【図5】

パケット単位でのリクエスト及びレスポンスの概念図である。

【図6】

(a)、(b)はprecinct data-binを説明するための図である。

【図7】

本発明の第1の実施形態に係るエンコード条件で符号化された画像の符号化データ、即ちJPEG2000のビットストリームの構成を示す図である。

【図8】

各タイルに付けられた番号を説明するための図である。

【図9】

本発明の第1の実施形態に係るキャッシュファイルを作成する為の処理のフロ

ーチャートである。

【図 10】

本発明の第 1 の実施形態に係るキャッシュファイルの概略構成を示す図である。

【図 11】

(a) は、初期化された状態の管理情報を示す図で、(b) は更新された管理情報を示す図である。

【図 12】

クライアント端末装置 201 がサーバ装置 204 に画像の符号化データを要求し、要求に応じてサーバ装置 204 がパケット毎に送信する画像の符号化データを受信し、受信した画像の符号化データをデコードして表示装置としての出力デバイス 105 に表示するまでの処理のフローチャートである。

【図 13】

サーバ装置 204 からパケット単位で受信した画像の符号化データをキャッシュする処理の詳細のフローチャートである。

【図 14】

配列 Info の構成を示す図である。

【図 15】

ステップ S1306 における処理の詳細を示すフローチャートである。

【図 16】

ステップ S1507, ステップ S1508 におけるキャッシュファイルの更新処理の詳細を示すフローチャートである。

【図 17】

キャッシュファイルから、各タイルのパケットのデータを出力ファイルに読み出す処理のフローチャートである。

【図 18】

本発明の第 3 の実施形態に係るキャッシュファイルの更新処理のフローチャートである。

【図 19】

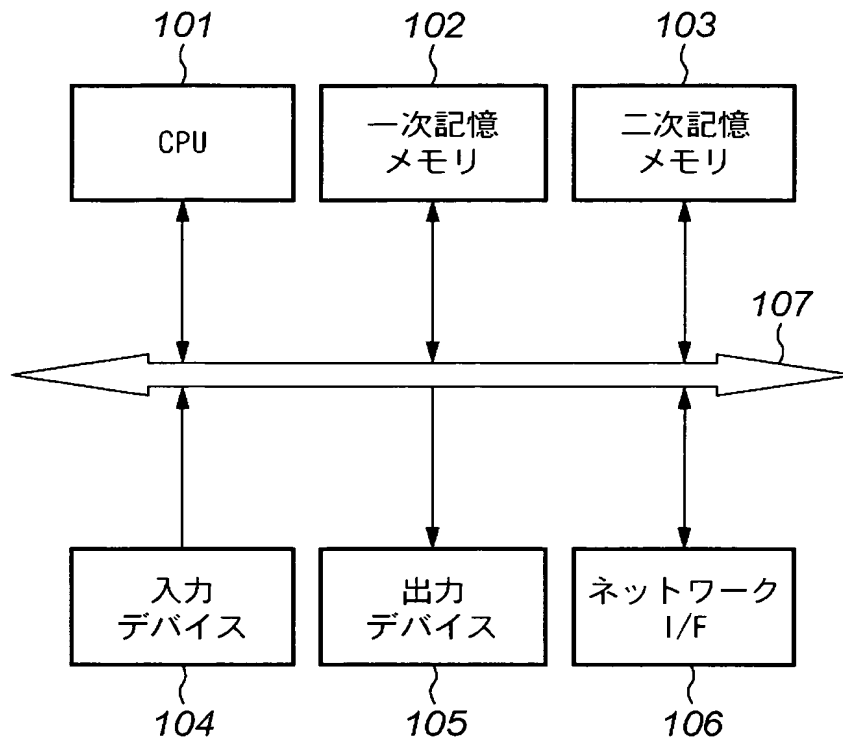
(a) は本発明の第 1 の実施形態に係るキャッシュファイルの作成処理によって作成されたキャッシュファイルの構成を示す図で、(b) は本発明の第 3 の実施形態に係るキャッシュファイルの作成処理によって作成されたキャッシュファイルの構成を示す図である。

【図 2 0】

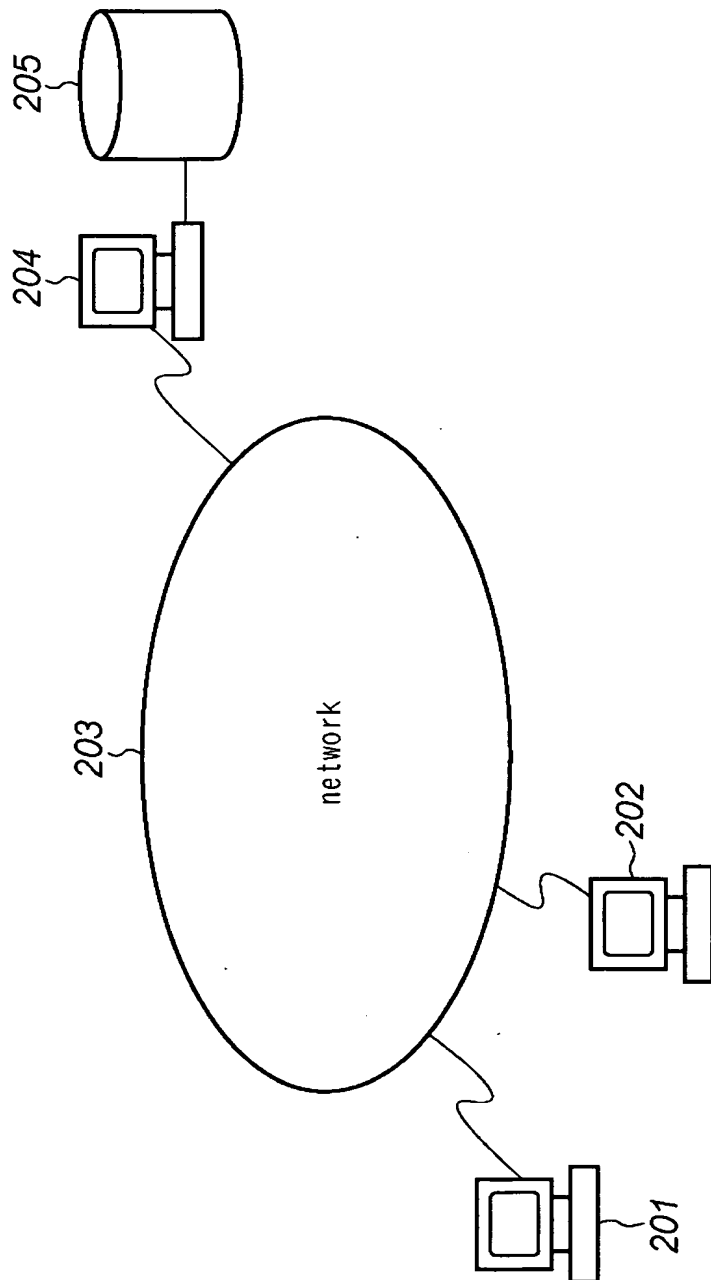
キャッシュファイルから、各タイルのパケットのデータを出力ファイルに読み出す処理のフローチャートである。

【書類名】 図面

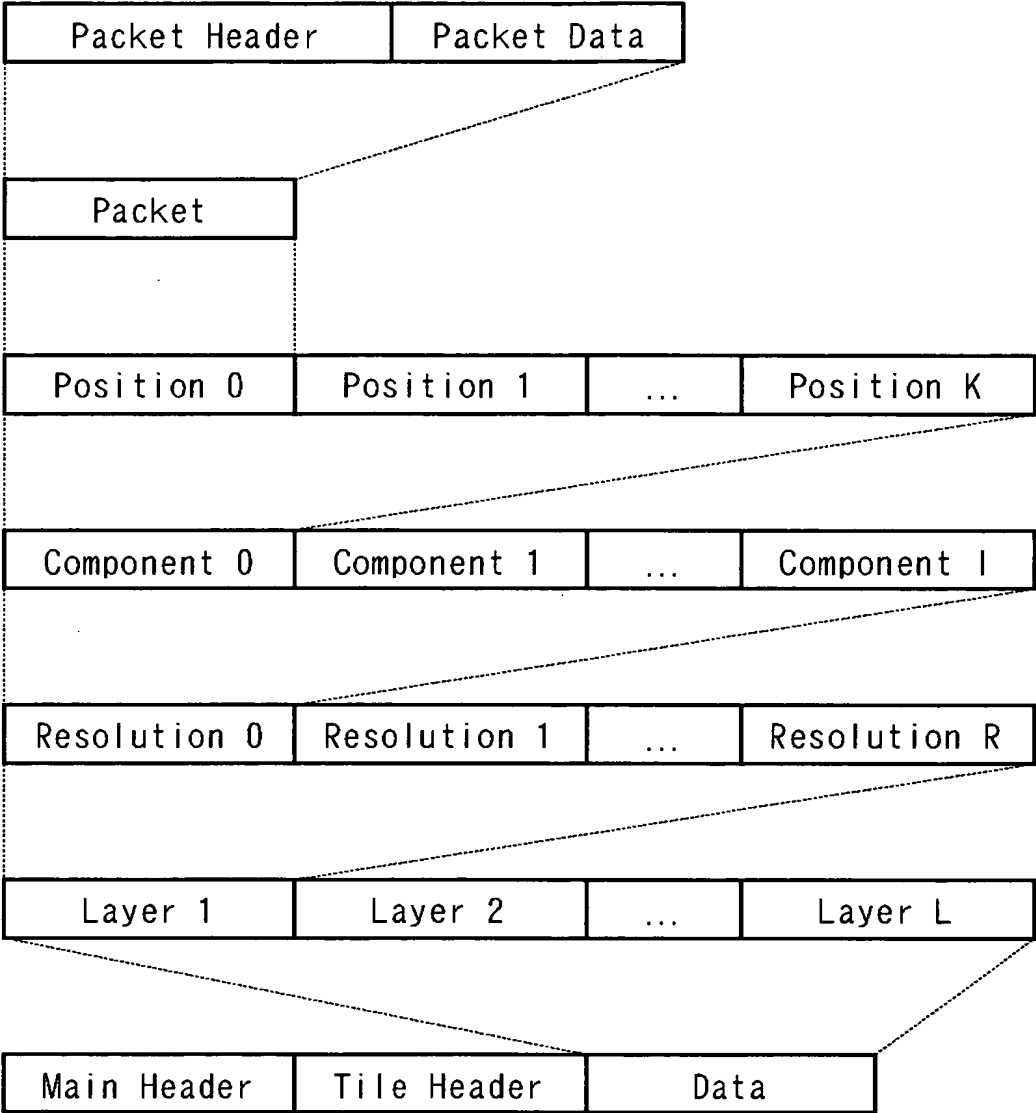
【図 1】



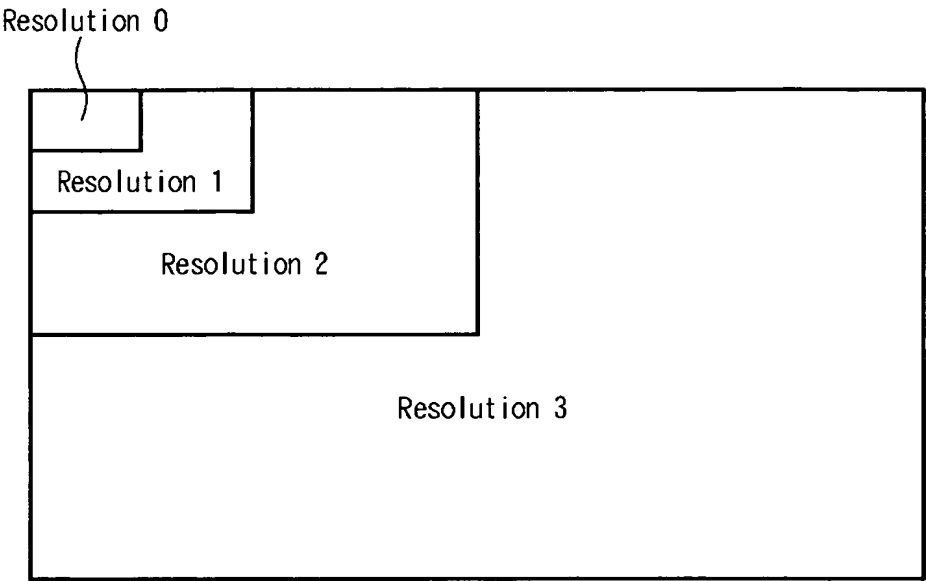
【図 2】



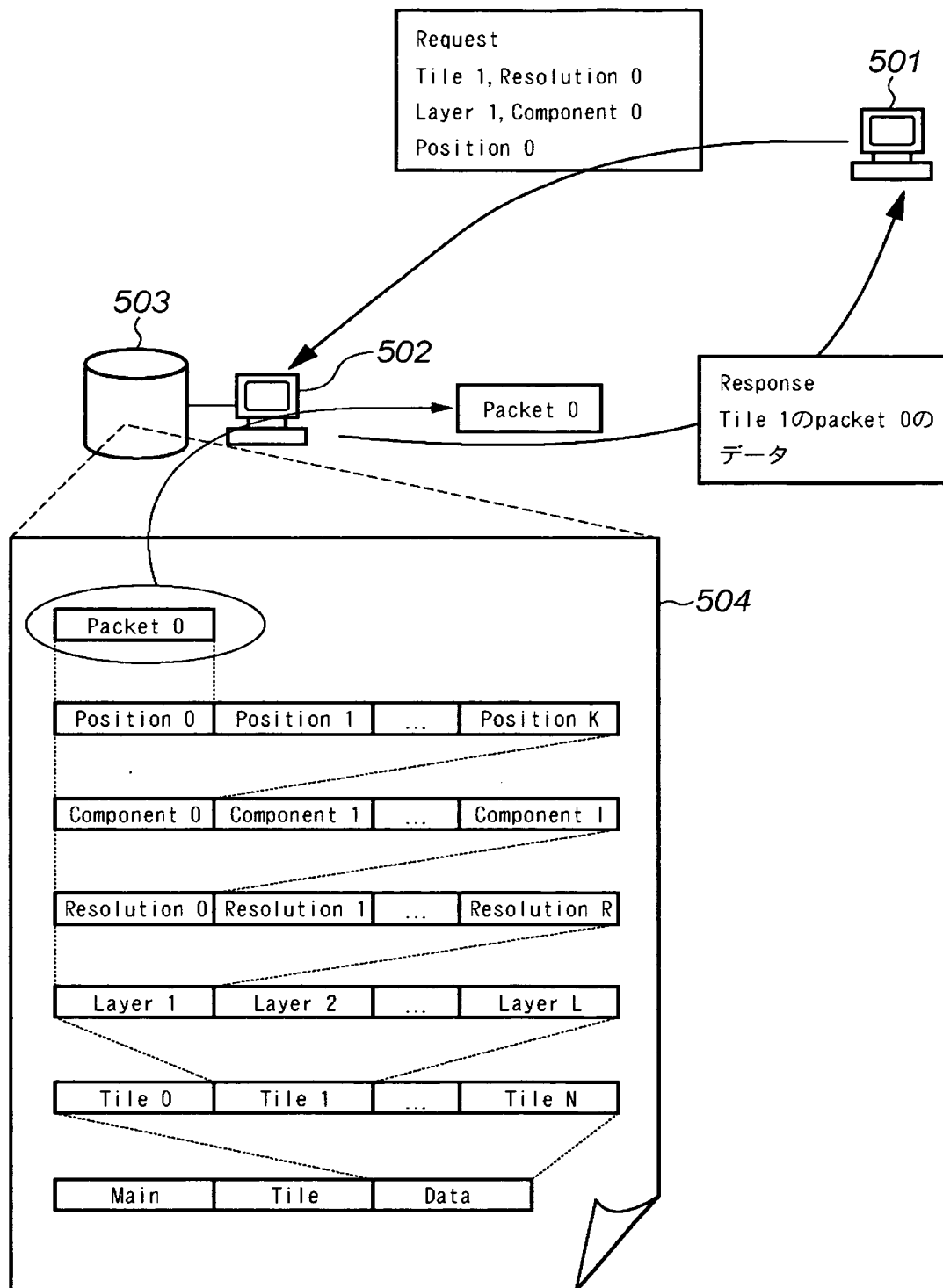
【図 3】



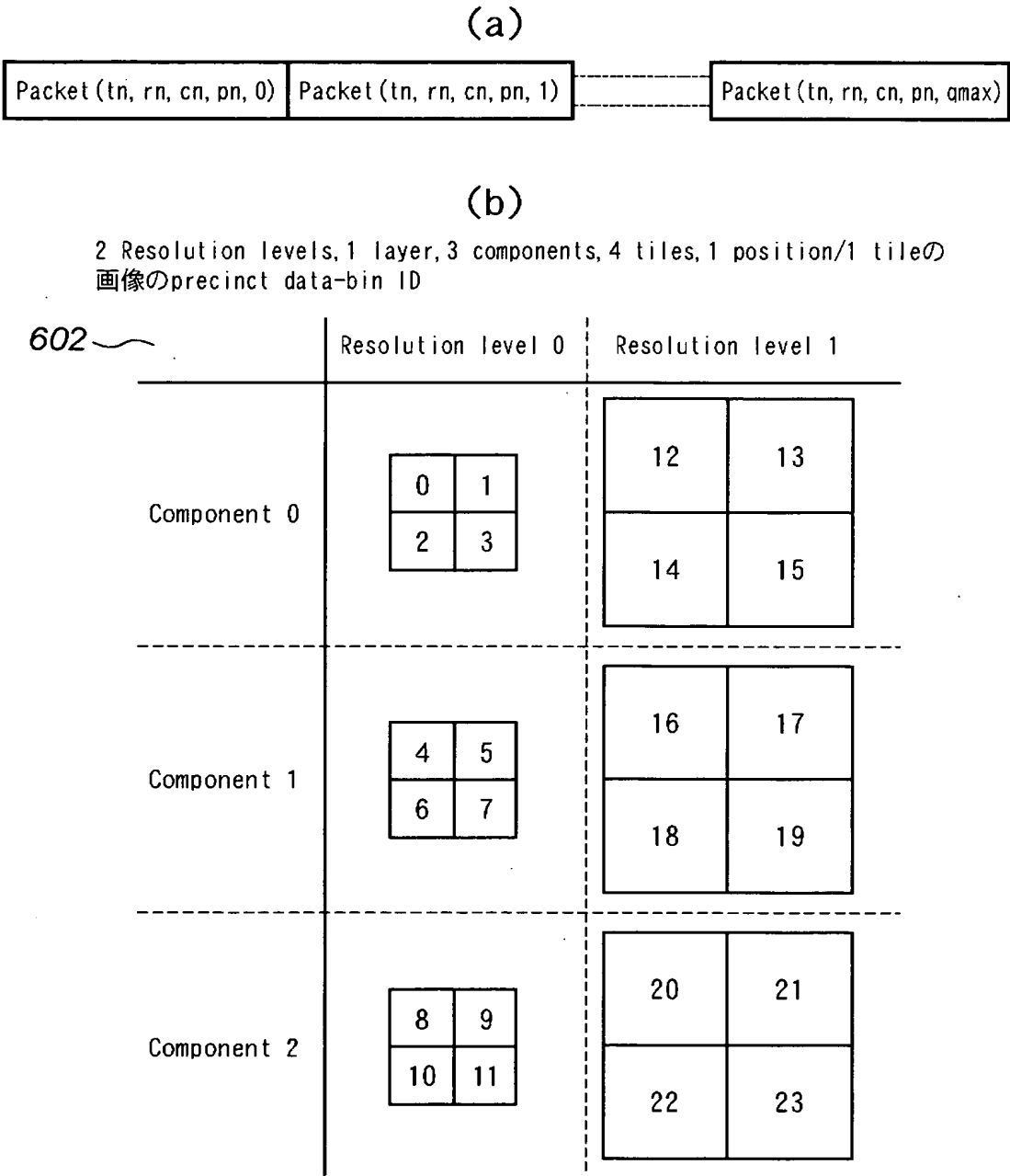
【図 4】



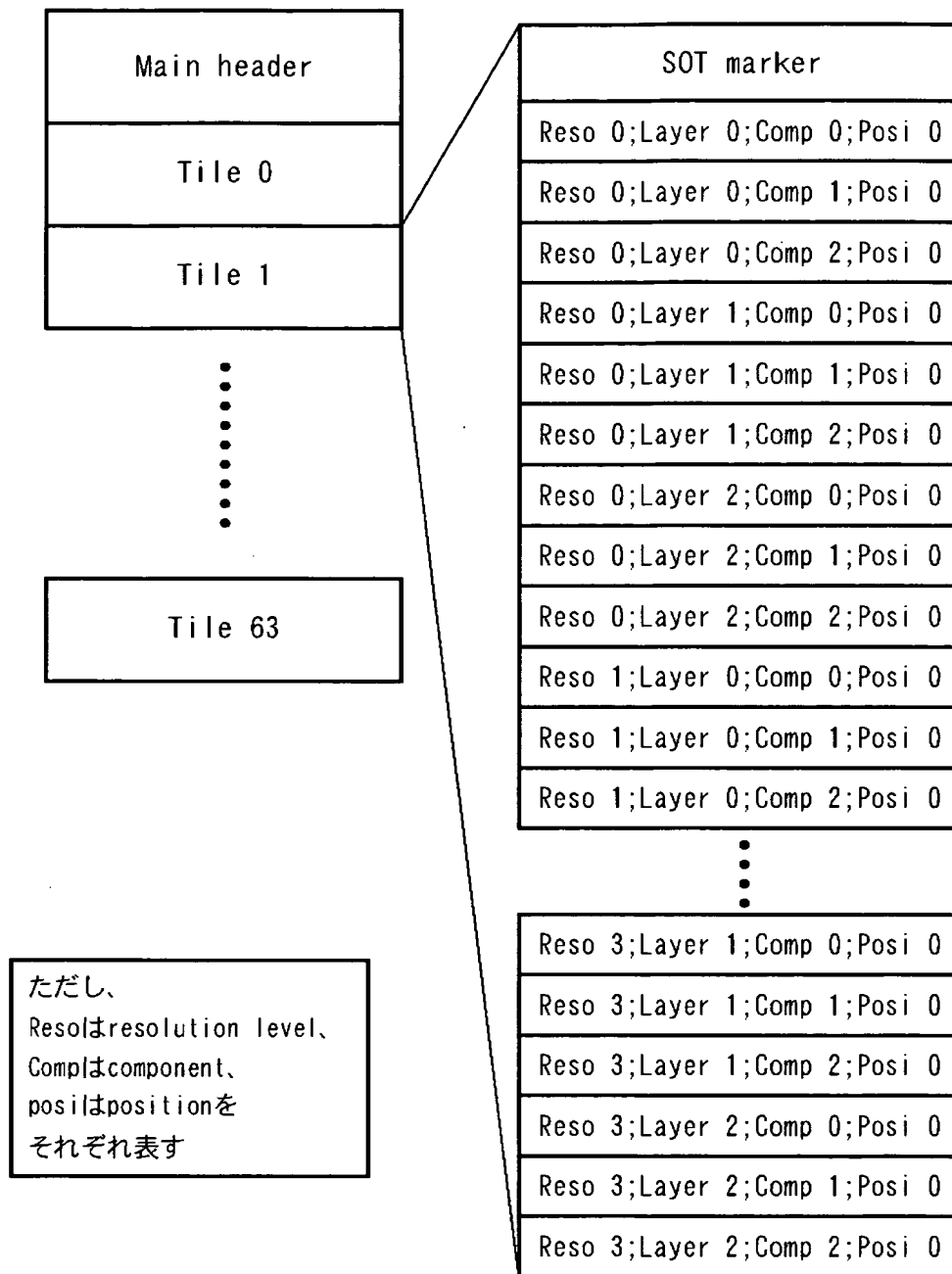
【図 5】



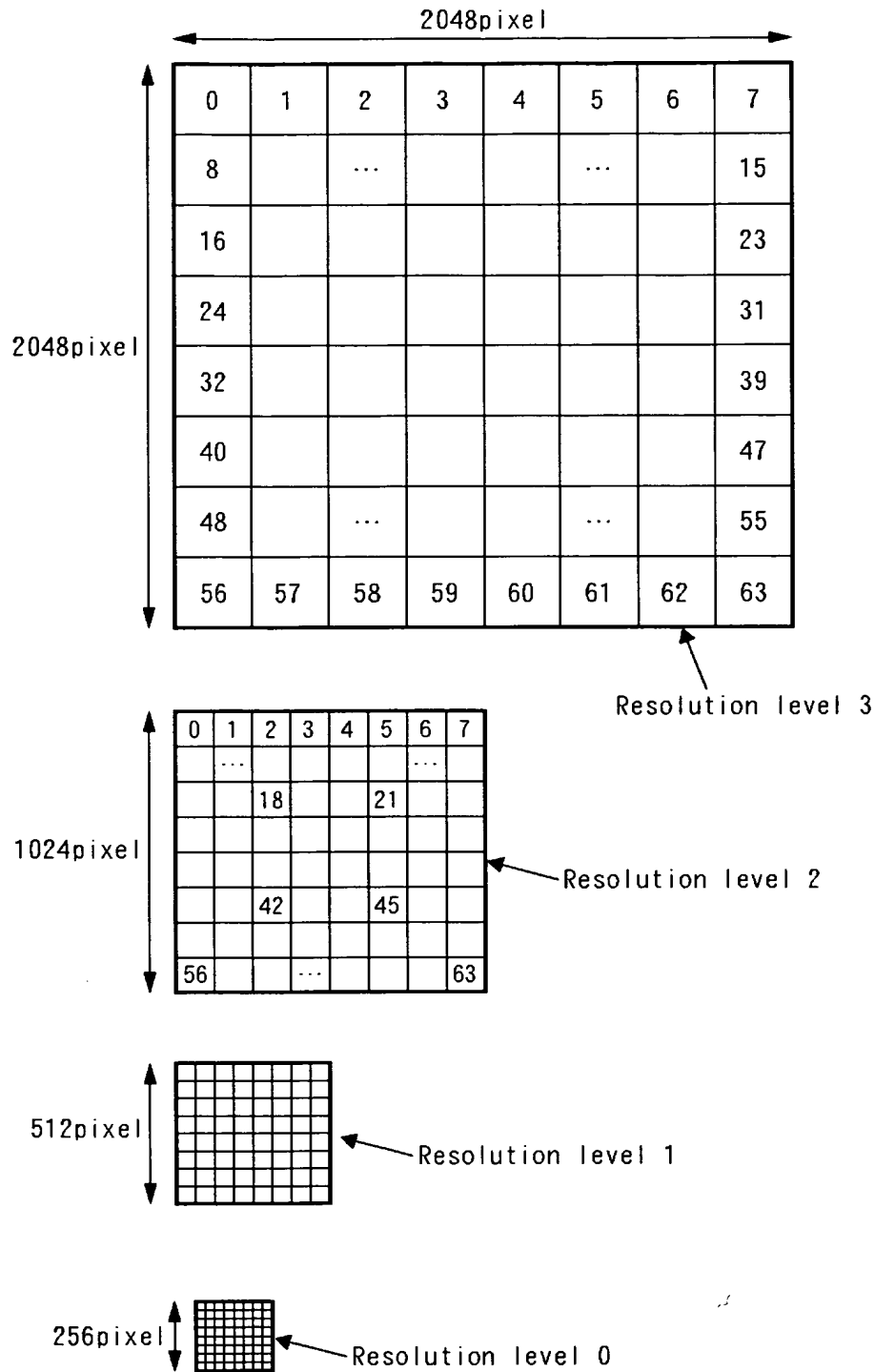
【図 6】



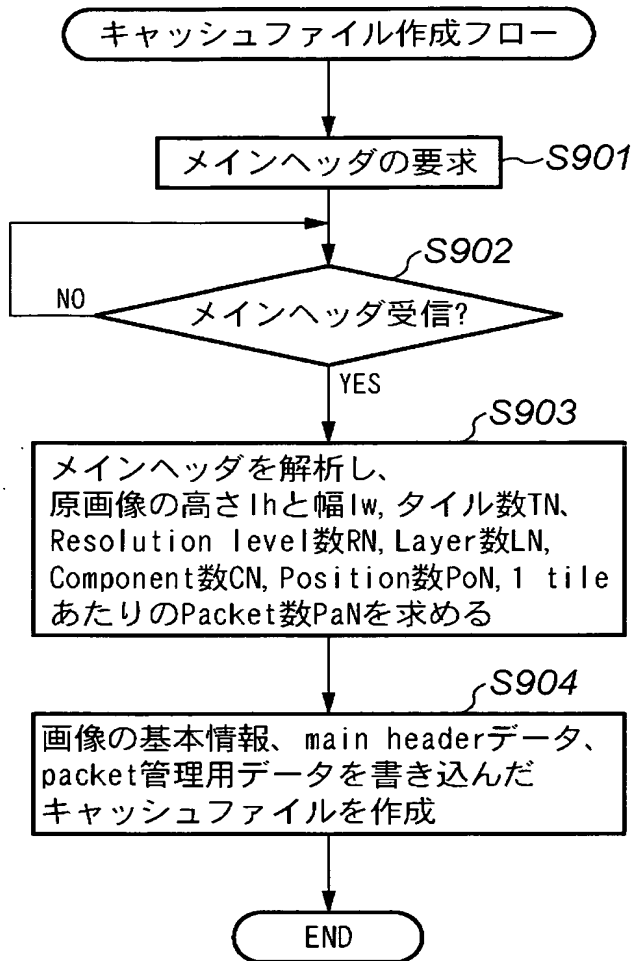
【図 7】



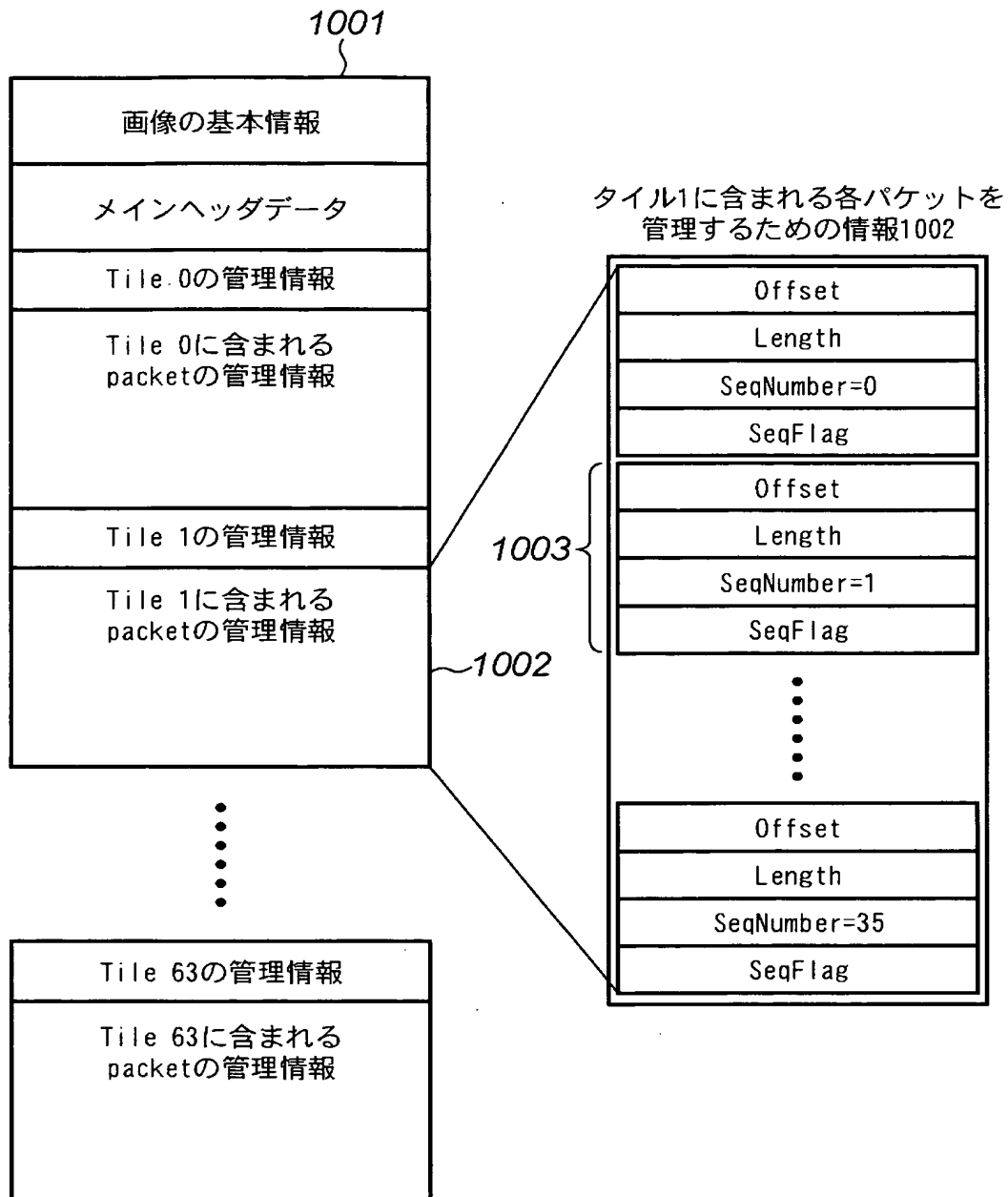
【図 8】



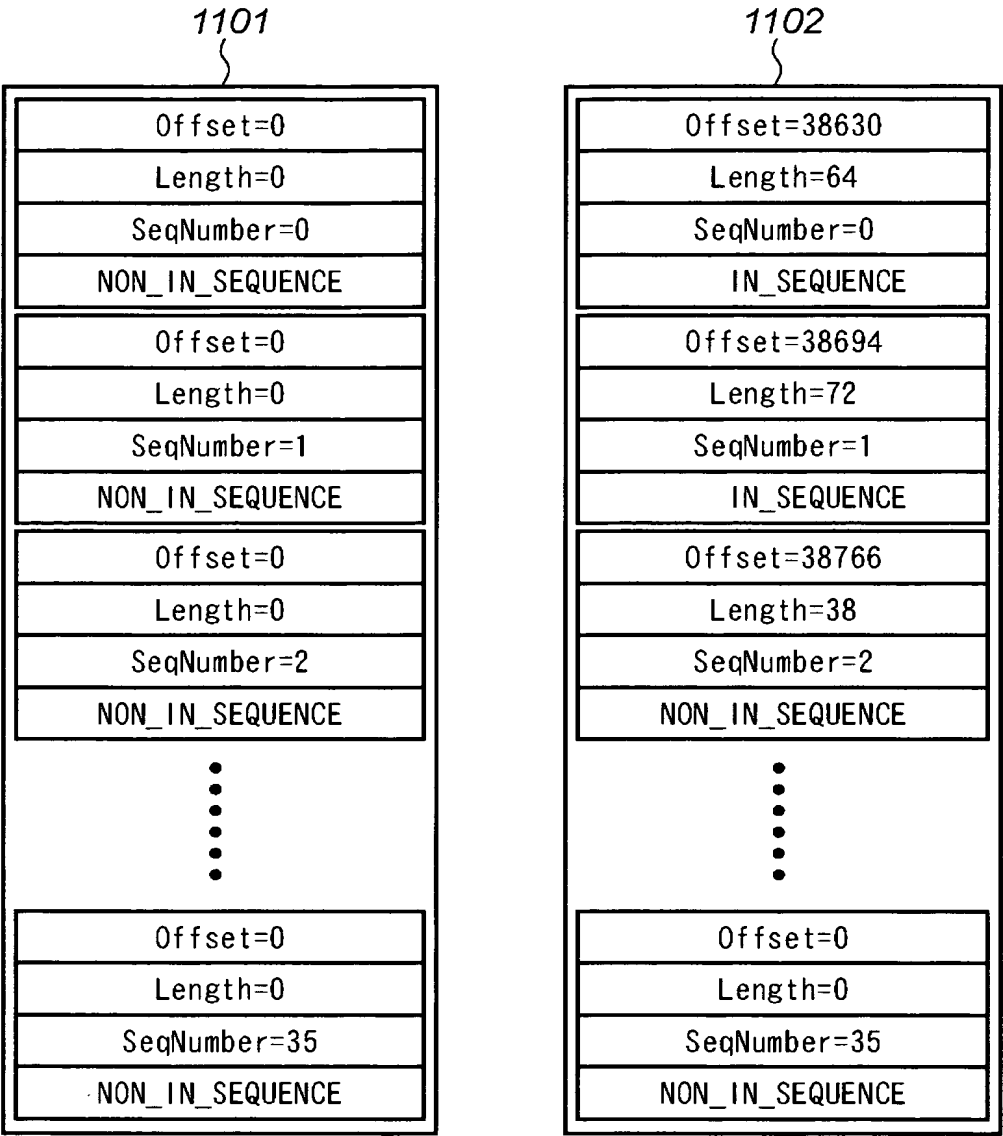
【図 9】



【図10】



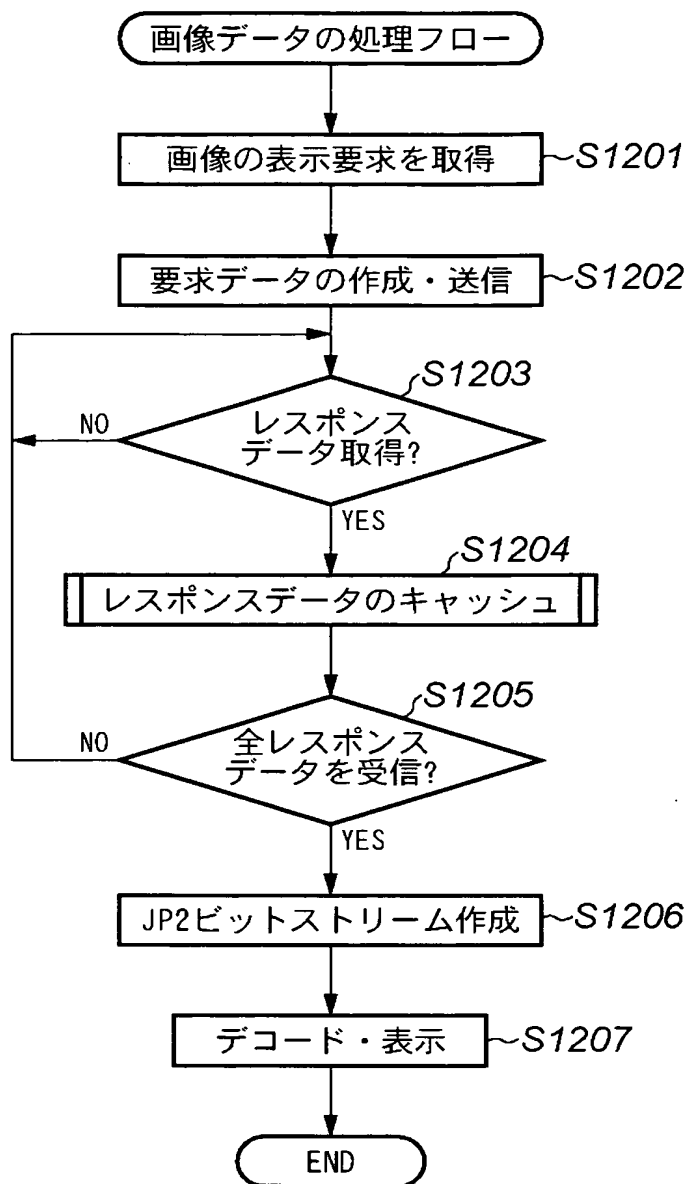
【図 11】



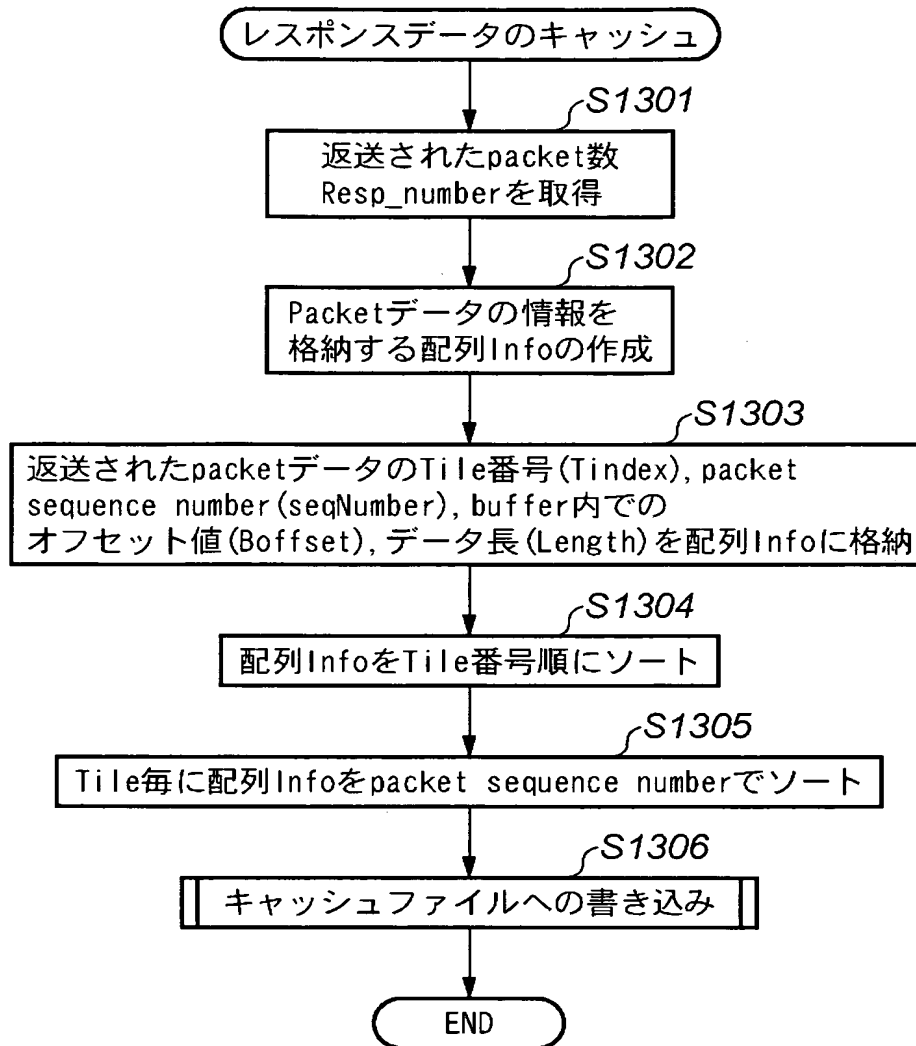
(a)

(b)

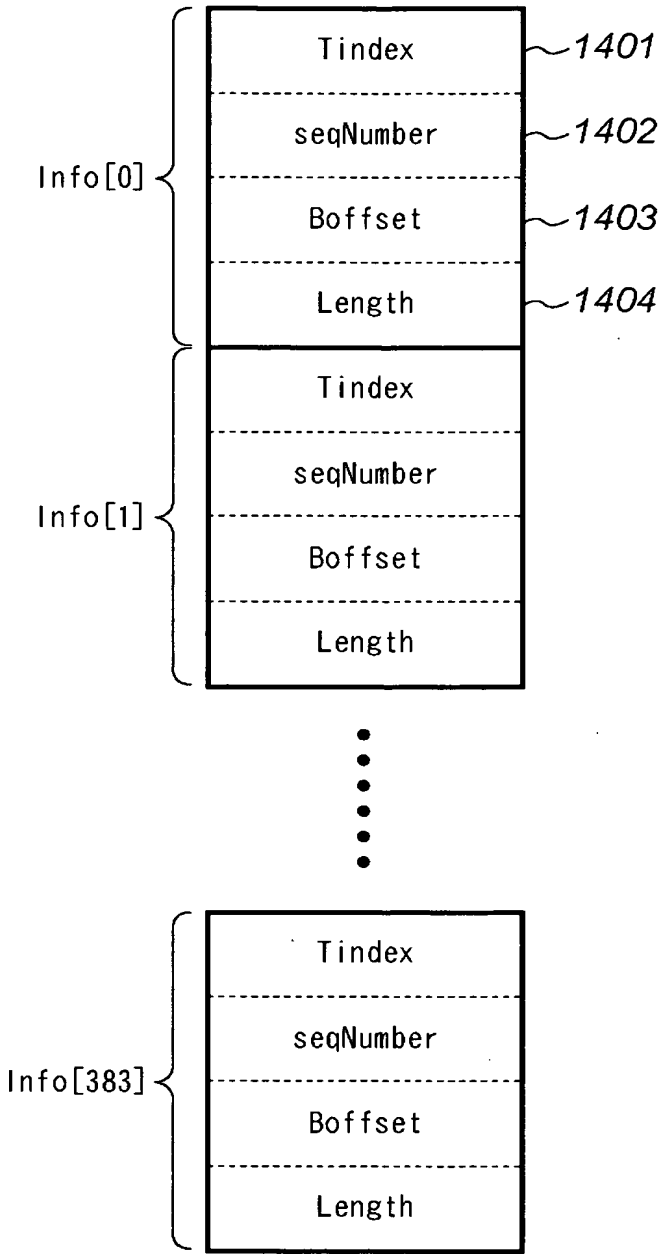
【図 12】



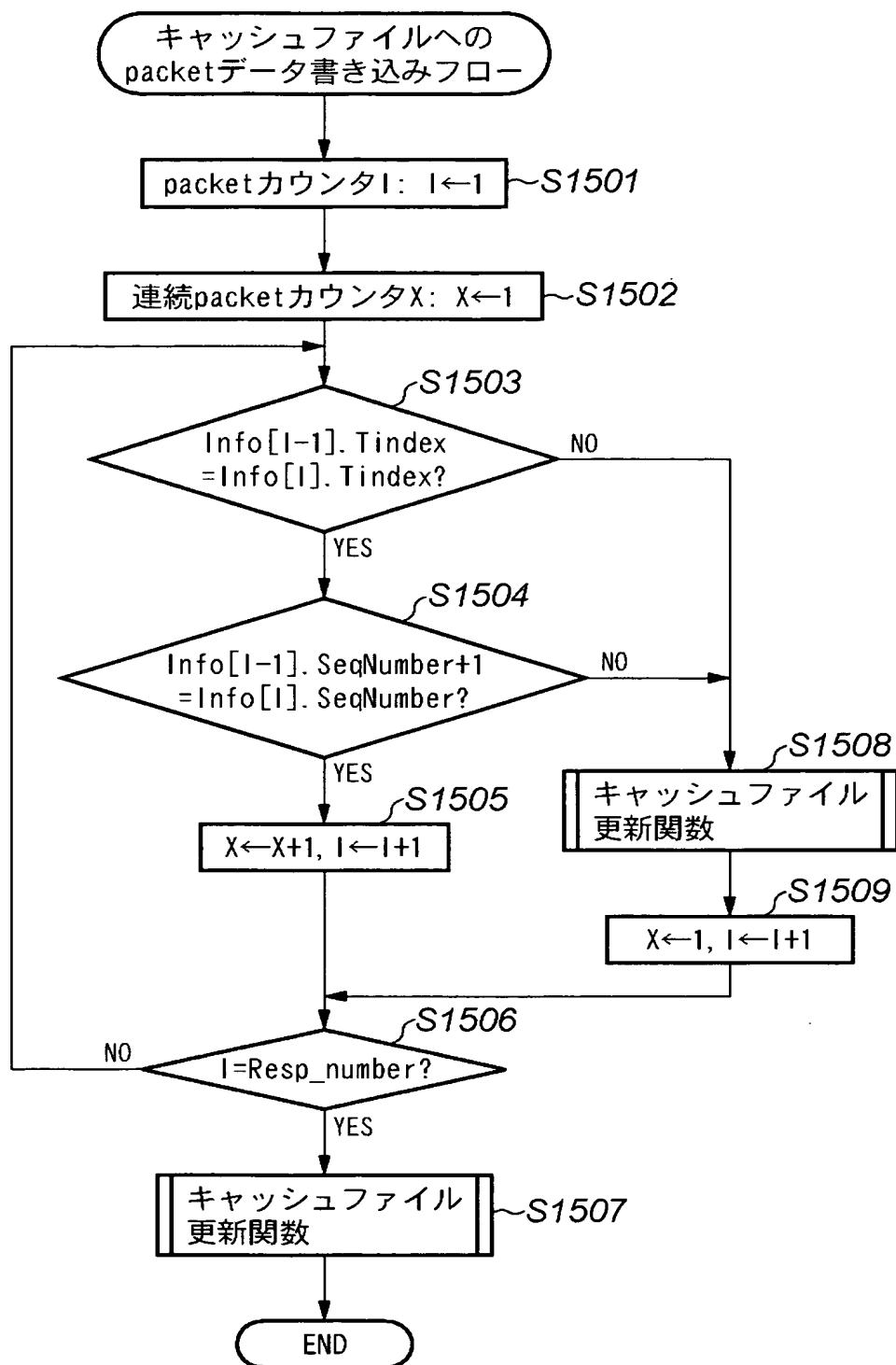
【図 13】



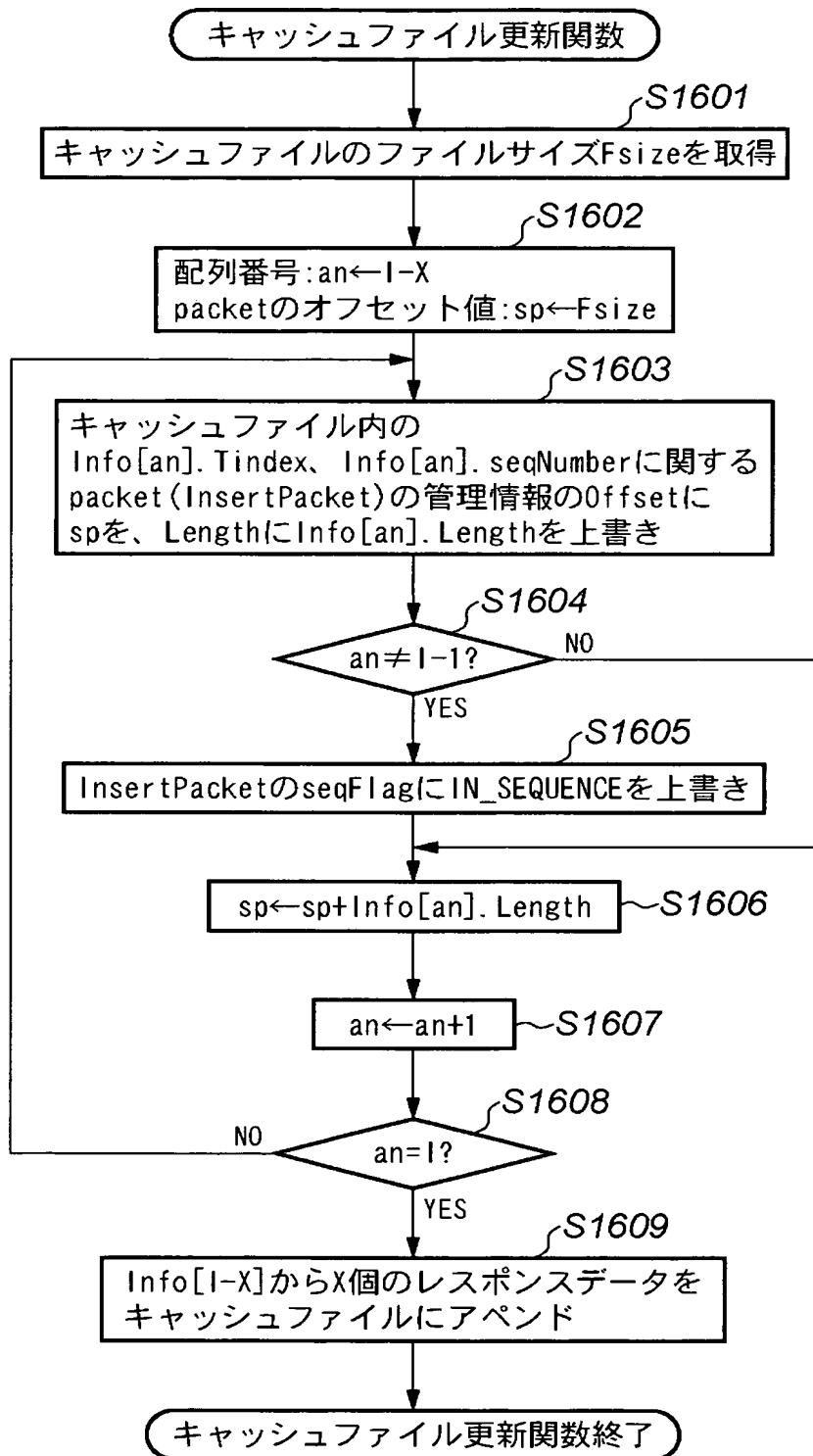
【図 1 4】



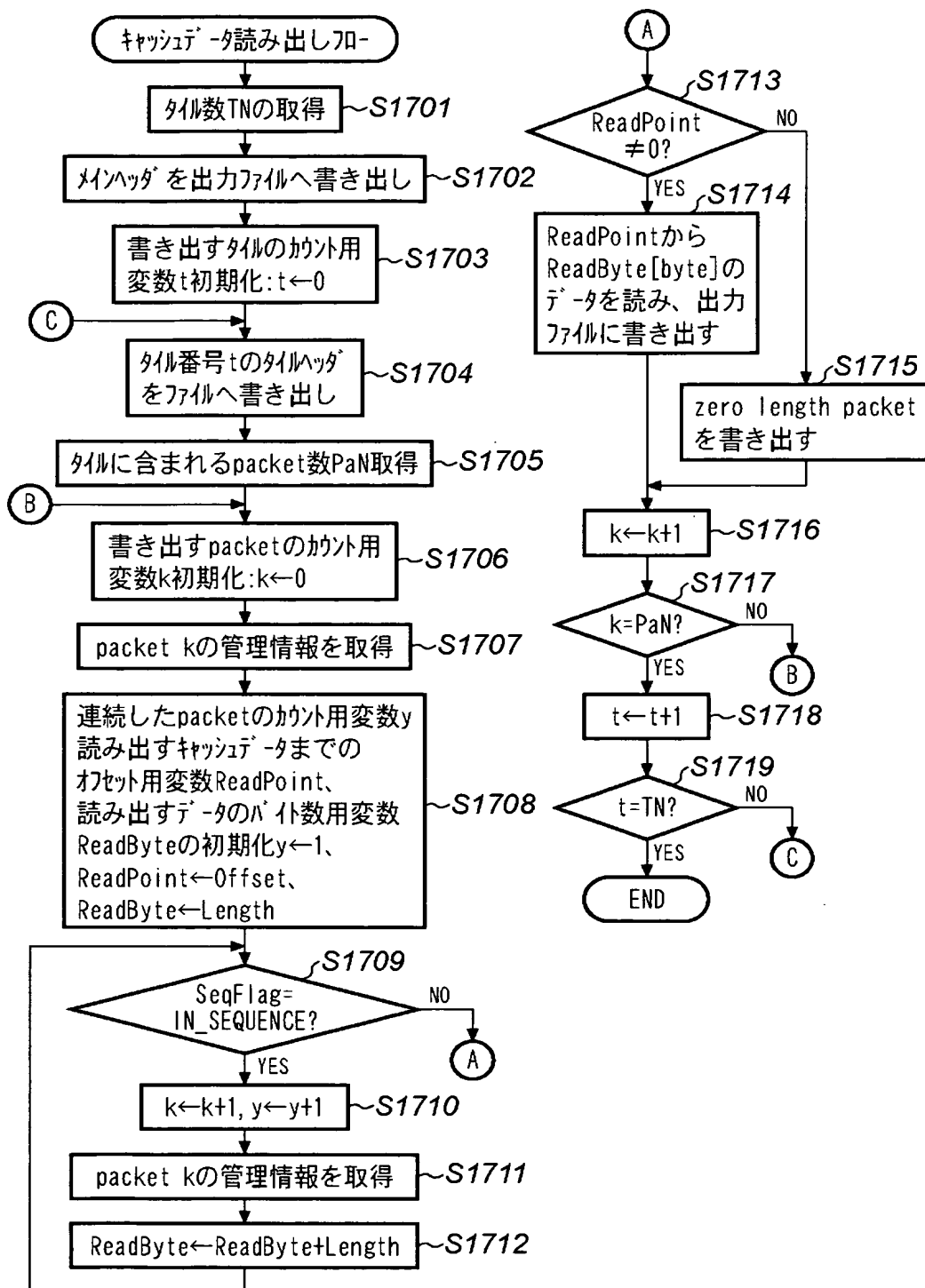
【図 15】



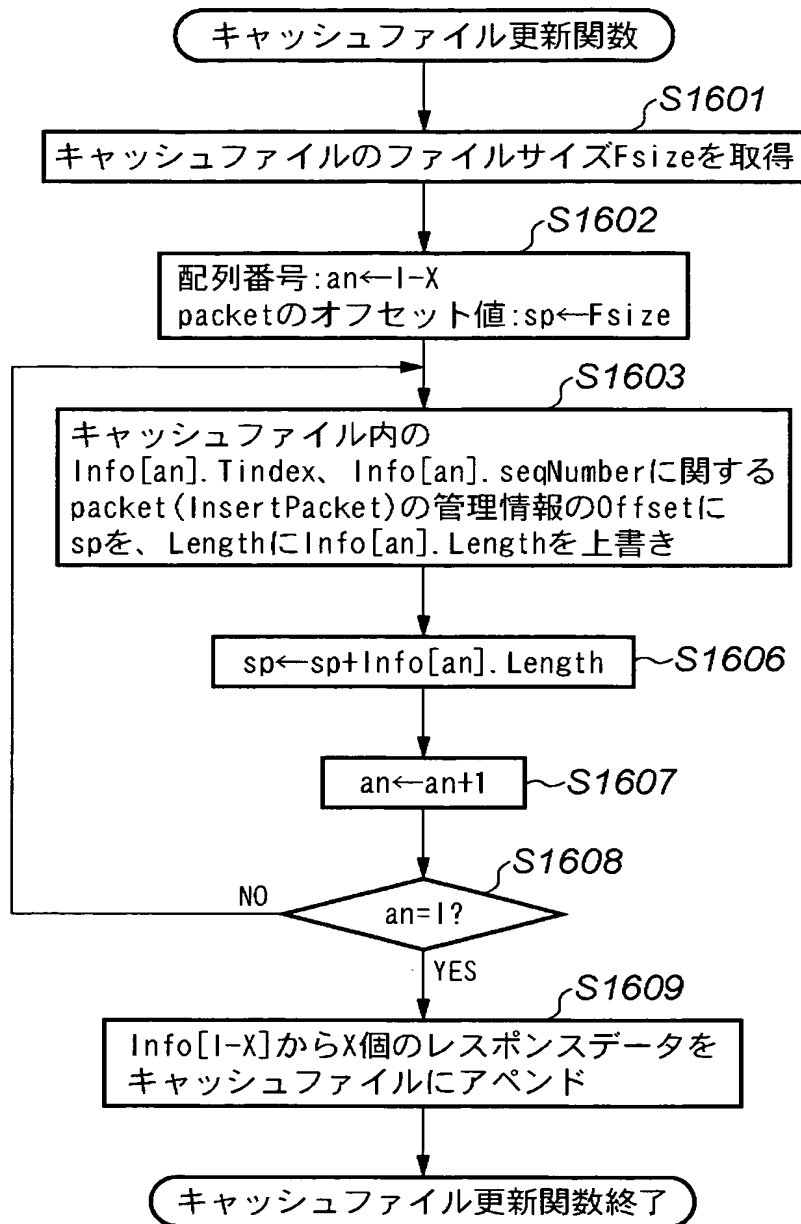
【図 16】



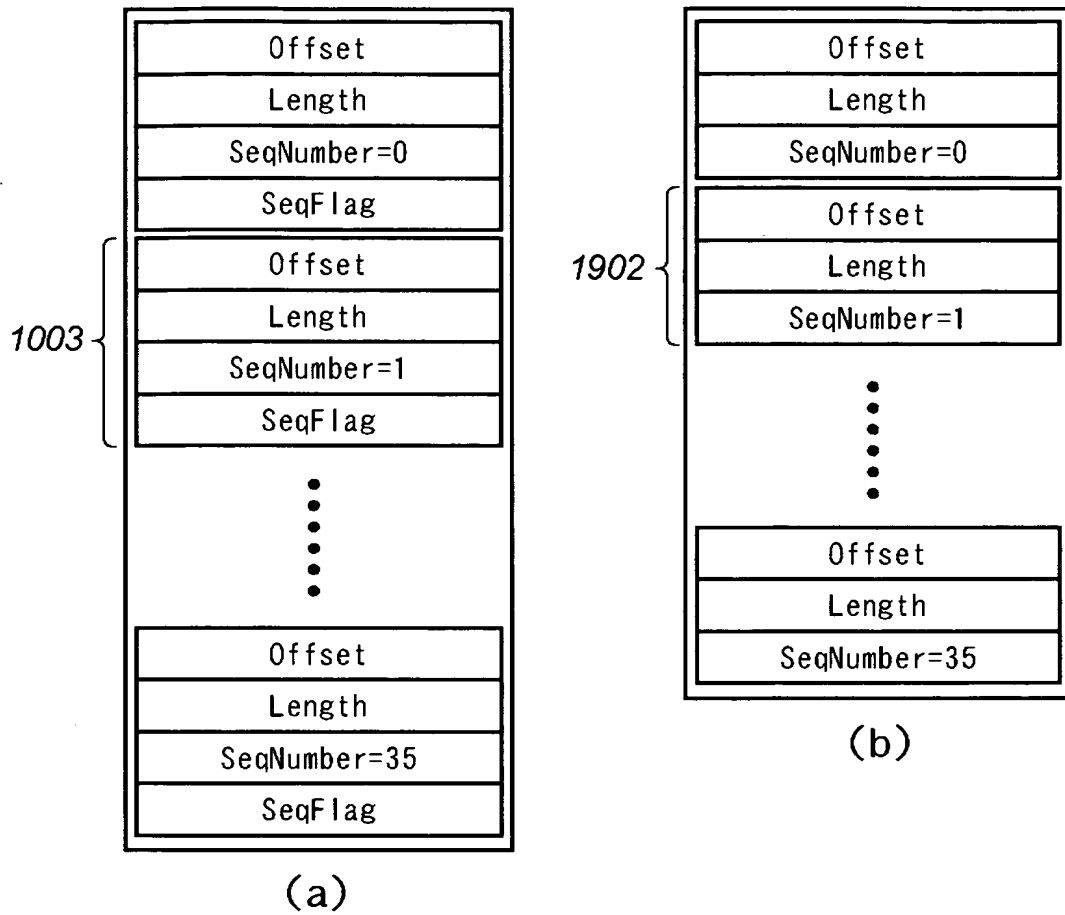
【図 17】



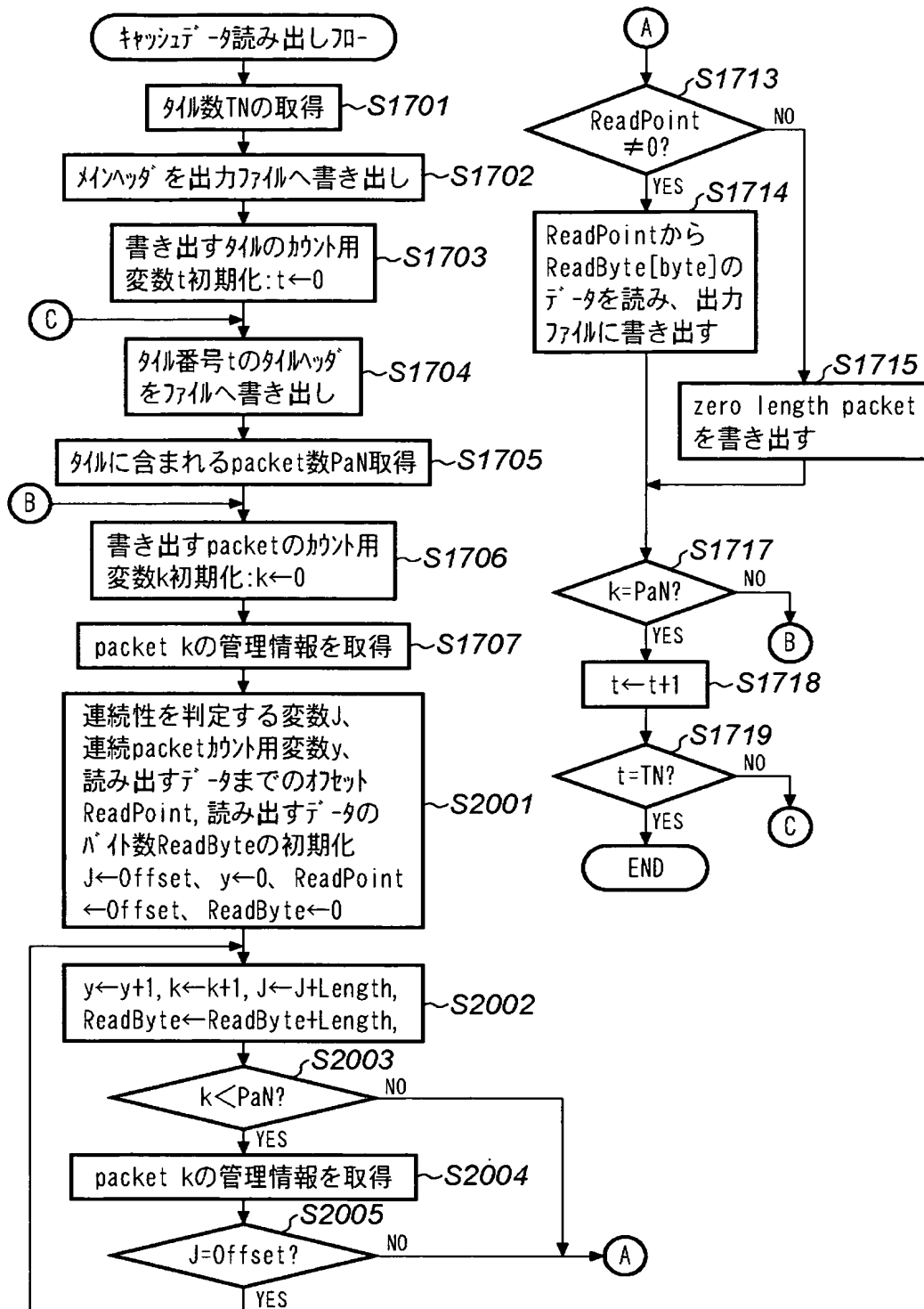
【図 18】



【図 1 9】



【図 20】



【書類名】 要約書

【要約】

【課題】 符号化データを受信してキャッシュする場合に、キャッシュしたデータに対してアクセスする際に要する時間や回数を軽減させること。

【解決手段】 複数のパケットから成るタイル単位で分割された画像の符号化データを保持する装置から、各タイルについて所望の画像を得るために必要な数のパケットのデータを受信する場合に、受信する各パケットを管理するための管理情報を作成し、受信した各々のパケットを、夫々のパケットが属するタイルの番号に従って並び替え（S1304）、更に、同じタイルに属する夫々のパケットをこのタイル中の順番に従って並び替え（S1305）、並び替えたパケットのデータを管理情報に順次付加すると共に、付加したパケットのデータと管理情報とを含むキャッシュデータ中の、各パケットのデータの配置に関する情報を管理情報に登録する（S1306）。

【選択図】 図13

特願 2 0 0 3 - 1 7 6 9 3 0

出 願 人 履 歴 情 報

識別番号

[0 0 0 0 0 1 0 0 7]

1 . 変更年月日

1 9 9 0 年 8 月 3 0 日

[変更理由]

新規登録

住 所

東京都大田区下丸子 3 丁目 3 0 番 2 号

氏 名

キヤノン株式会社